



**Universidad**  
**Zaragoza**

## Trabajo Fin de Grado

Operadores software para el filtrado de imágenes con  
luz polarizada

Software operators for image filtering with polarized  
light

Autor

Fernando del Molino Bello

Director

Adolfo Muñoz Orbañanos

ESCUELA DE INGENIERÍA Y ARQUITECTURA  
2018







(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. Fernando del Molino Bello

con nº de DNI 73009197K en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)  
Grado, (Título del Trabajo)  
Operadores software para el filtrado de imágenes con luz polarizada

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 20 de Noviembre de 2018

Fdo: Fernando del Molino

# RESUMEN

Este documento recoge el procedimiento que se ha seguido para implementar una aplicación para aplicar filtros polarizadores en fase de post-proceso utilizando los vectores de Stokes.

Se ha realizado un estudio de los vectores de Stokes y de las matrices de Mueller, herramientas que permiten representar el estado de polarización de la luz y manipularlo, para posteriormente, utilizar estos conceptos para desarrollar algoritmos de filtrado. Estos algoritmos permiten tanto aplicar filtros polarizadores convencionales, como un polarizador lineal o circular, como filtros para optimizar parámetros como el brillo o el contraste que son imposibles de obtener en la realidad. Estos algoritmos se desarrollan de forma óptima para minimizar el tiempo de ejecución, permitiendo que posteriormente se usen para una aplicación en tiempo real.

También se introduce el método de captura de imágenes necesario para generar estas imágenes de Stokes, con la ventaja que al contrario que otros métodos, solo necesita una cámara convencional y dos filtros polarizadores.

Por último, se introduce una herramienta software que permite, mediante el uso de pinceles, editar una imagen manipulando los parámetros anteriormente introducidos, como el brillo o la saturación, utilizando los filtros comentados anteriormente.

# Índice

<b>1. Introducción y objetivos</b>	<b>1</b>
1.1. Objetivos . . . . .	3
1.2. Trabajo previo . . . . .	4
1.3. Estructura . . . . .	4
<b>2. Polarización</b>	<b>6</b>
2.1. Tipos de polarización . . . . .	7
2.2. Elipse de polarización . . . . .	9
2.3. Vectores de Stokes . . . . .	10
2.4. Mueller . . . . .	12
2.4.1. Polarizador lineal . . . . .	13
2.4.2. Retardador . . . . .	14
2.4.3. Rotador . . . . .	15
2.4.4. Combinaciones de elementos . . . . .	16
<b>3. Filtros polarizadores en post-proceso</b>	<b>17</b>
3.1. Imágenes de Stokes . . . . .	17
3.2. Captura de imágenes . . . . .	18
3.3. Filtrado software . . . . .	21
3.3.1. Filtro de polarización lineal . . . . .	22
3.3.2. Filtro circular . . . . .	22
3.3.3. Filtro parametrizado . . . . .	23
3.4. Filtros de optimización . . . . .	25
3.4.1. Filtros globales . . . . .	26
3.4.2. Filtros locales . . . . .	32
<b>4. Edición con pinceles</b>	<b>41</b>
<b>5. Conclusiones</b>	<b>47</b>
<b>Bibliografía</b>	<b>49</b>

<b>Lista de Figuras</b>	<b>51</b>
<b>Anexos</b>	<b>54</b>
<b>A. Elipse de polarización</b>	<b>55</b>
<b>B. Elementos polarizadores</b>	<b>58</b>
B.1. Polarizador lineal . . . . .	58
B.2. Retardador . . . . .	60
B.3. Retardador y combinación de elementos . . . . .	62
<b>C. Derivaciones</b>	<b>63</b>
C.1. Contraste . . . . .	63
C.2. Saturación . . . . .	64

# Capítulo 1

## Introducción y objetivos

En las ultimas dos décadas, el campo de la fotografía ha sufrido grandes avances. Aunque la fotografía digital ya existía en 1975, no fue hasta el siglo XXI cuando esta se introdujo en la sociedad. Con la llegada de las cámaras digitales, se han generado nuevos campos, como la fotografía computacional, que une el campo de la fotografía con la informática. En los últimos años se ha visto el auge de tecnologías como el HDR (High Dynamic Range), que entre otras cosas introduce un nuevo formato de imagen, que permite realizar operaciones por software como el mapeado de tonos. Otra tecnología introducida en este milenio son las cámaras plenópticas, como la cámara Lytro, que captura información sobre los *lightfield* (campos de luz), introduciendo un formato de imagen que permite, de nuevo a posteriori a la toma de la fotografía, cambiar la distancia focal y la profundidad de campo. Ambas tecnologías se pueden ver en la Figura 1.1.



(a) Ejemplo del uso de la tecnologia HDR (b) Ejemplo del cambio de enfoque a posteriori de para mejorar el contraste de una imagen. una imagen tomada con una cámara plenóptica.

Figura 1.1: Distintas tecnologías que permiten una edición a posteriori

En fotografía, es muy común el uso de **filtros polarizadores**. La luz, que es una onda electromagnética, oscila en dos campos, el eléctrico y el magnético, que oscilan en un plano perpendicular a la dirección de propagación de la onda en todas las direcciones posibles. Un filtro polarizador limita las direcciones en la que estos campos pueden oscilar. La mayoría de fotógrafos cuentan con alguno de estos en su equipo debido a su utilidad. Los filtros polarizadores permiten modificar los colores que se

captan en una imagen, modificando el contraste o la saturación de la imagen. Este se utiliza por ejemplo, para intensificar los cielos azules, como se puede ver en la Figura 1.2. Otra de las funciones que permiten realizar los filtros polarizadores es modificar los reflejos sobre el agua o sobre otro tipo de superficies.



(a) Izquierda, foto sin polarizador. Derecha, foto con polarizador lineal.



(b) Izquierda, foto sin polarizador. Derecha, foto con polarizador lineal.

Figura 1.2: Comparativa de la misma fotografía tomada con y sin polarizador donde se observan los efectos que se pueden conseguir con estos.

La principal idea de este trabajo es, al igual que en las tecnologías comentadas anteriormente, que permiten modificar algunos efectos o parámetros en fase de post-proceso de la imagen, aplicar este mismo concepto a las imágenes con filtros polarizadores. Para ello va a ser necesario estudiar como se comportan estos filtros polarizadores, para lo que es necesario comprender el comportamiento de la luz al interactuar con este tipo de filtros. Además se introducirá un formato de imagen, que de manera análoga al HDR y a los *lightfields* permita aplicar a posteriori filtros polarizadores (ver Figura 1.3).

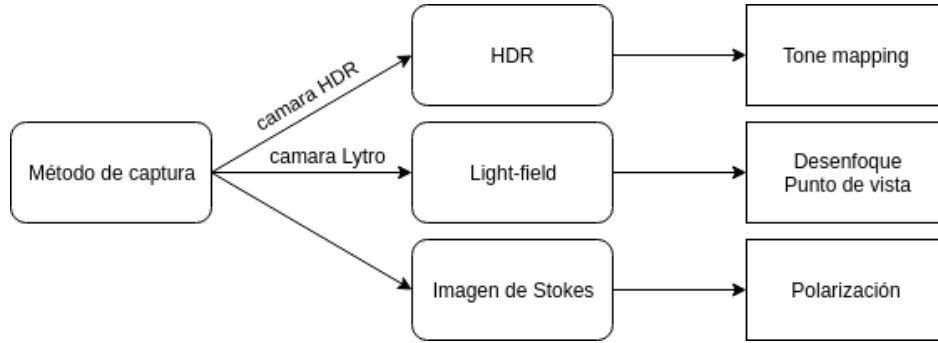


Figura 1.3: Esquema de las analogías entre las tecnologías HDR, lytro y el objetivo a conseguir mediante la representación de Stokes.

Para la representación de ondas electromagnéticas polarizadas, el método mas utilizado son los vectores de Stokes, introducidos por George Gabriel Stokes en 1852 [1]. Posteriormente en 1943 Hans Mueller estableció el cálculo que recibe su propio nombre, cálculo de Mueller [2], que permite manipular los vectores de Stokes, así como representar los elementos básicos de la polarización como los polarizadores lineales o los retardadores.

## 1.1. Objetivos

Los objetivos de este trabajo están divididos en cuatro fases claramente diferenciadas, siendo los dos últimos el principal aporte de este trabajo:

1. Estudiar el comportamiento de la polarización de la luz, su representación mediante los **parámetros de Stokes**, y la manipulación de estos últimos con el **cálculo de Mueller**.
2. Capturar imágenes de Stokes y desarrollar los algoritmos necesarios para aplicar filtros a estas imágenes.
3. Estudiar el comportamiento de la **luminosidad**(brillo), el **contraste** y la **saturación** de una imagen polarizada, y utilizar este estudio para desarrollar filtros capaces de optimizar estos tres parámetros. Estos filtros serán de dos tipos, globales, en los que se aplicara el mismo filtro a toda la imagen, y locales, donde se aplicaran filtros distintos en cada pixel de la imagen. Son especialmente interesantes estos últimos ya que con ellos se obtienen resultados imposibles de conseguir mediante la fotografía convencional. Además, estos algoritmos de optimización se desarrollarán teniendo en cuenta el tiempo de ejecución, para posteriormente utilizarlos en una aplicación en tiempo real.



4. Desarrollar una herramienta que permita editar los parámetros comentados anteriormente, luminosidad, contraste y saturación, mediante el uso de **pinceles**, permitiendo maximizar y minimizar estos parámetros en zonas concretas de la imagen.

## 1.2. Trabajo previo

Aunque el tratamiento de imágenes polarizadas por software no ha sido muy explotado, cabe destacar el Trabajo de Fin de Master de David Osantes, "Fotografía computacional para el filtrado de la polarización en fase de post-proceso" [3]. En él, se desarrolla la idea del filtrado de polarización por software, por un lado aplicar filtros polarizadores a elección del usuario, y por otro aplicar filtros por objetivos, para maximizar o minimizar el brillo o el contraste. Sin embargo, la selección de estos filtros es limitada, ya que se parte de una preselección de filtros, y se selecciona el mejor mediante prueba y error. Además carece de un filtrado local, lo que impide aplicar un filtrado distinto en diferentes zonas de la imagen.

En cuanto a la explotación de las propiedades de los filtros polarizadores en imagen computacional y edición de imágenes no hay mucho trabajo al respecto. Existen trabajos que utilizan la captura de imágenes polarizadas para posteriormente eliminar los brillos especulares de diferentes superficies [4], pero no se centran en la posibilidad de edición fotográfica.

Por otro lado, el uso de los vectores de Stokes esta ampliamente extendido en distintas disciplinas, por lo que es interesante explorar el campo de los vectores de Stokes ya que tiene otras muchas aplicaciones. Estas aplicaciones van desde la óptica, las comunicaciones, el estudio de tejidos biológicos [6], y en este trabajo, la fotografía.

## 1.3. Estructura

El capítulo 2 contienen el conocimiento teórico necesario para llevar a cabo los distintos objetivos del trabajo. En primer lugar se trata la polarización de la luz, introduciendo las principales características de este efecto que sufren las ondas electromagnéticas. Además, se hablará sobre las distintas formas de clasificar las ondas polarizadas. Posteriormente, se realizará un estudio sobre los vectores de Stokes, una de las herramientas básicas para la realización de este trabajo, donde se explicará como construirlos y como manipularlos. En esta última parte se introducirán las



matrices de Mueller, que representan los operadores básicos para trabajar con los vectores de Stokes.

Los capítulos 3 y 4 recogen el núcleo del trabajo. En el capítulo 3 se desarrollarán los distintos algoritmos utilizados para el filtrado de imágenes en postproceso, tanto de los filtros comunes en fotografía, como el filtro lineal y el circular, como de los filtrados para maximización y minimización de parámetros, tanto globales como locales, así como una comparación entre estos. En el capítulo 4, se utilizarán los filtrados explicados en el capítulo anterior para generar una herramienta que permita manipular estos parámetros mediante el uso de pinceles sobre la imagen.

Por último, el capítulo 5 incluye las conclusiones finales del trabajo y también los posibles puntos para explorar en un futuro.

# Capítulo 2

## Polarización

En este capítulo se va a introducir el concepto de polarización, así como las propiedades de este fenómeno. Es importante conocer las propiedades de la polarización y su comportamiento ya que todo el trabajo posterior se asienta sobre estos conceptos.

La **polarización** es una propiedad de las ondas transversales, aquellas cuyas oscilaciones se producen perpendiculares a la dirección de propagación de la onda, entre las que se encuentra la luz. Esta es una onda electromagnética, y en estas, tanto el campo eléctrico como el magnético oscilan en diferentes direcciones, siendo ambas perpendiculares entre si y a su vez perpendiculares a la dirección de propagación de la onda. El plano en el que oscilan estos campos se denomina plano de polarización, donde por convención, se utiliza el campo eléctrico para referirse a la polarización de una onda. Este plano se define con dos vectores, ambos perpendiculares a la dirección de propagación. En una onda electromagnética no polarizada, el campo eléctrico oscila en todas las direcciones perpendiculares a la dirección de propagación de la onda, mientras que en una onda polarizada, las direcciones del campo están limitadas en algún aspecto.

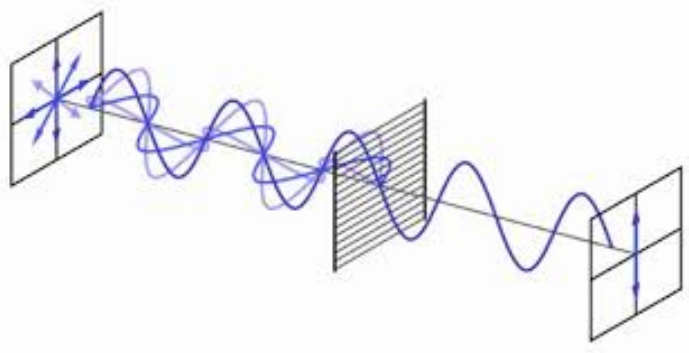


Figura 2.1: Polarización de una onda. La onda no polarizada oscila en todas las direcciones posibles del plano de polarización, al atravesar el elemento polarizador, oscila solo en una dirección.

El estado de polarización de una onda se define a partir de la evolución temporal que sufre el campo eléctrico de dicha onda en un punto concreto del espacio, y en un plano perpendicular a la dirección de propagación (plano de polarización).

Si se considera una onda propagándose en la dirección  $z$  de un sistema de coordenadas ortogonal  $xyz$ , el campo eléctrico se puede representar como la suma de dos campos perpendiculares que se propagan en la dirección normal al plano  $xy$  (la dirección  $z$ ):

$$\begin{aligned} E_x(z, t) &= E_{ox} \cos(Kz - \omega t + \delta_x) \\ E_y(z, t) &= E_{oy} \cos(Kz - \omega t + \delta_y) \end{aligned} \quad (2.1)$$

- $E_{ox}$  y  $E_{oy}$  son las amplitudes del campo eléctrico en los ejes de referencia  $x$  e  $y$  respectivamente.
- $K$  representa el número de ondas, que podemos obtener conociendo la longitud de onda en el vacío, ya que  $K = \frac{2\pi}{\lambda}$  siendo  $\lambda$  la longitud de onda.
- $\omega$  es la frecuencia angular, que se obtiene de la frecuencia natural  $f$ ,  $\omega = 2\pi f$ .
- $\delta_x$  y  $\delta_y$  representa la diferencia de fase de ambas componentes.

Si se fija este plano en un punto  $z$  cualquiera del espacio y se define el cambio de fase  $\delta = \delta_x - \delta_y$ :

$$\begin{aligned} E_x(t) &= E_{ox} \cos(\omega t) \\ E_y(t) &= E_{oy} \cos(\omega t + \delta) \end{aligned} \quad (2.2)$$

Se puede observar que estas ecuaciones son las ecuaciones paramétricas de una elipse, luego la evolución temporal de estas componentes define una elipse llamada **elipse de polarización**, que se utiliza entre otras cosas para categorizar los distintos tipos de polarización.

## 2.1. Tipos de polarización

La luz se puede clasificar en tres grandes grupos en función del grado de polarización que hayan sufrido:

- **Luz no polarizada:** es aquella en la que no existe ningún predominio de ninguna elipse de polarización. La mayoría de la luz natural es de este tipo.
- **Luz polarizada totalmente:** se produce cuando a lo largo del tiempo de medida, la elipse de polarización mantiene su forma y los ángulos de elipticidad y acimut son constantes

- **Luz polarizada parcialmente:** se produce cuando a lo largo del tiempo de medida hay un predominio de algún tipo de elipse. A efectos prácticos se puede considerar una superposición de un haz no polarizado y otro totalmente polarizado

Además de esta clasificación, se pueden realizar otras clasificaciones en función del valor de  $\delta$ , ya que además de una elipse, en valores concretos de  $\delta$ , la evolución temporal del campo eléctrico puede generar un círculo o una línea (ver Figura 2.2)

- Si  $\delta = n \cdot \pi$  para  $n=0,1,2,..$  **Polarización lineal:** en este caso la evolución del campo eléctrico genera una recta.
- Si  $\delta = \frac{\pi}{2} + n \cdot \pi$  para  $n=0,1,2,..$  y  $E_{ox} = E_{oy}$  **Polarización circular:** la evolución del campo eléctrico genera una circunferencia.
- El resto de casos se conocen como **polarización elíptica**

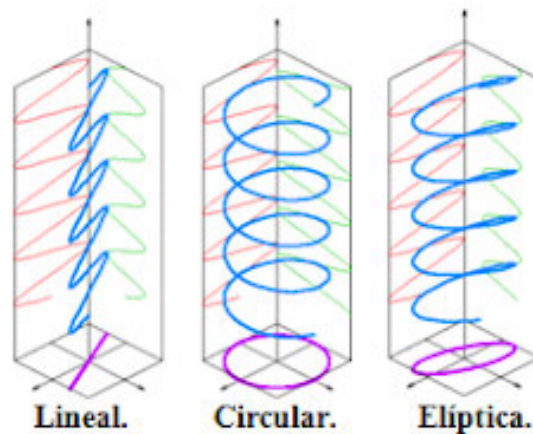


Figura 2.2: Ilustración de los diferentes tipos de polarización en función de la forma que generan en el plano de polarización.

Se puede realizar otra clasificación en función del sentido de giro de la polarización de la onda:

- Si  $0 < \delta < \pi$  se denomina polarización elíptica levógira o helicidad negativa. El sentido de giro de la polarización es en el sentido de las agujas del reloj.
- Si  $\pi < \delta < 2\pi$  se denomina polarización elíptica dextrógira o helicidad positiva. El sentido de giro de la polarización es en el sentido contrario a las agujas del reloj.

Como se puede ver en la Figura 2.3, la primera fila corresponde a una polarización levógira mientras que la segunda se corresponde con una polarización dextrógira.

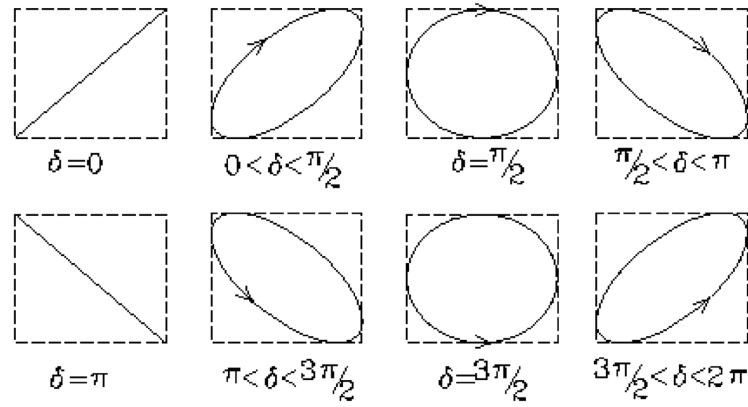


Figura 2.3: Cambios en la elipse de polarización en función de la diferencia de fase  $\delta$

## 2.2. Elipse de polarización

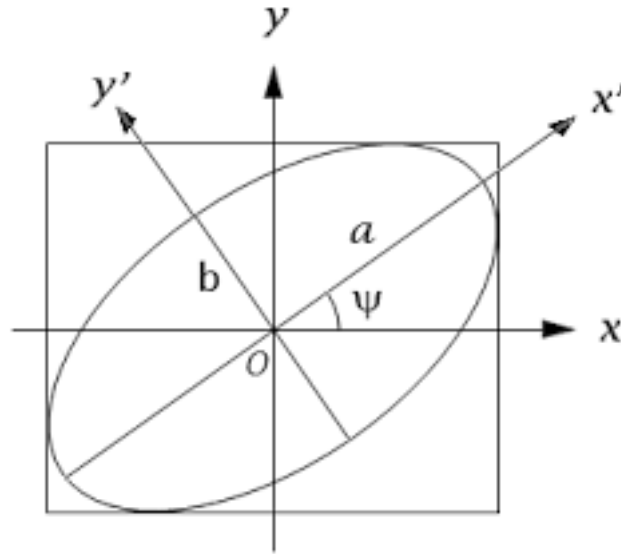


Figura 2.4: Elipse de polarización sobre los ejes de referencia  $x$  e  $y$ .

En la Figura 2.4 se pueden observar los distintos parámetros que aparecen en la elipse de polarización:

- $O$ ,  $x$  e  $y$  representan el sistema de referencia.
- $x'$  e  $y'$  representan los ejes de la elipse.
- $\Psi$  representa el azimut, la inclinación de los ejes de la elipse en el plano de polarización. Si el  $\Psi = 0$ ,  $x' = x$  e  $y' = y$ .

- b y a representan el semieje menor y mayor de la elipse respectivamente. Sirven para medir la elipticidad de la elipse, definida por  $\chi = \frac{b}{a}$

En el anexo A se incluye la demostración de que las ecuaciones del campo representan una elipse, así como se incluyen más propiedades de la misma.

## 2.3. Vectores de Stokes

En las secciones anteriores se han introducido el concepto de polarización así como sus propiedades. El siguiente paso es poder trabajar con este concepto, generar filtros, manipular la elipse de polarización, etc. Para poder realizar estas operaciones, en primer lugar se necesita una forma de representar el estado de polarización de la luz, y posteriormente un método para trabajar con esta representación. En primer lugar se va a tratar la representación de la polarización, los vectores de Stokes, y posteriormente se abordará la forma de manipularlos, con el método de Mueller (sección 2.4).

Los **vectores de Stokes** son una estructura que sirve para representar el estado de polarización de la luz. Fueron definidos por George Gabriel Stokes en 1852 [1]. Estos vectores están compuestos por cuatro componentes distintas que permiten representar todos los **estados de polarización** posibles. Las componentes son las siguientes:

$$\begin{aligned} S_0 &= E_{ox}^2 + E_{oy}^2 \\ S_1 &= E_{ox}^2 - E_{oy}^2 \\ S_2 &= 2E_{ox}E_{oy} \cos \delta \\ S_3 &= 2E_{ox}E_{oy} \sin \delta \end{aligned} \tag{2.3}$$

Estos parámetros se agrupan de la siguiente forma, y a continuación se va a explicar el significado de cada uno de ellos:

$$\begin{pmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{pmatrix} = \begin{pmatrix} I \\ Q \\ U \\ V \end{pmatrix} \tag{2.4}$$

El primer parámetro,  $S_0$  o  $I$ , representa la intensidad total de la luz. El segundo,  $S_1$  o  $Q$ , indica la predominancia de la polarización horizontal ( $0^\circ$ ) sobre la polarización vertical ( $90^\circ$ ). El tercero,  $S_2$  o  $U$ , describe la predominancia de la polarización a  $45^\circ$  sobre la polarización a  $135^\circ$ , y por último, el cuarto parámetro  $S_3$  o  $V$  representa la predominancia de la polarización circular levógira (a derechas) sobre la polarización circular dextrógira (a izquierdas). Aunque se utilizan ambas notaciones,  $S_0$  o  $I$ , a partir

de este punto se va a utilizar la notación I, Q, U y V para referirse a cada uno de los parámetros.

Los parámetros de Stokes se pueden definir además en un función de dos haces polarizados:

$$\begin{aligned}
I &= E_{ox}^2 + E_{oy}^2 = I_{0^\circ} + I_{90^\circ} \\
Q &= E_{ox}^2 - E_{oy}^2 = I_{0^\circ} - I_{90^\circ} \\
U &= 2E_{ox}E_{oy} \cos \delta = I_{45^\circ} - I_{135^\circ} \\
V &= 2E_{ox}E_{oy} \sin \delta = I_{rcp} - I_{lcp}
\end{aligned}$$

- $E_{ox}$  y  $E_{oy}$  representan la amplitud de onda en los ejes de referencia X e Y respectivamente.
- $\delta$  es la diferencia de fase entre la componente X y la componente Y.
- $I_{x^\circ}$  representa un haz polarizado linealmente a x grados.
- $I_{rcp}$  e  $I_{lcp}$  representa un haz con una polarización circular a derechas y a izquierdas respectivamente (rcp y lcp por sus siglas en inglés, *right or left circular polarization*).

Al observar la definición de los parámetros, queda claro que el primer parámetro I tiene que ser un valor positivo, mientras que el resto oscilarán su valor entre 1 y -1, (teniendo en cuenta que  $E_{ox}$  y  $E_{oy}$  estén entre 0 y 1) en función de la polarización de ese haz de luz. A continuación se muestran los valores de I, Q, U y V para distintos tipos de polarización: para una polarización lineal a  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$ , y para una polarización circular a derechas (RCP) e izquierdas(LCP).

$0^\circ$	$90^\circ$	$45^\circ$	$135^\circ$	RCP	LCP
$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}$

Los casos que se muestran en la tabla anterior son vectores de Stokes para luz totalmente polarizada, sin embargo, los parámetros de Stokes permiten también representar luz parcialmente polarizada y luz natural (ver tipos de polarización en la Sección 2.1).

Para el caso de la luz natural (luz no polarizada), se puede observar que todos los parámetros menos I van a ser 0, ya que no hay predominancia de ningún tipo de

polarización sobre otra. Así, el vector de Stokes  $S$  para la luz natural sera:

$$S = I \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (2.5)$$

El otro caso que queda por tratar es el de la luz polarizada parcialmente. Este tipo de polarización se considera como una superposición de estados, por un lado un estado de luz polarizada totalmente y otro de luz no polarizada. Para medir la aportación de cada estado se define el parámetro  $G$ , **grado de polarización**, que indica que porcentaje del haz está polarizado. Se define mediante la ecuación 2.6 y tomara valor  $G=1$  siempre que la luz sea completamente polarizada, y por el contrario, tendrá valor  $G=0$  la luz no polarizada.

$$G = \frac{\text{polariada}}{\text{total}} = \frac{\sqrt{Q^2 + U^2 + V^2}}{I} \quad (2.6)$$

## 2.4. Mueller

Como se ha comentado al comienzo de la sección anterior, se necesitan dos cosas para manipular la elipse de polarización y las propiedades de esta. El primer elemento necesario es una forma de representar la polarización, problema solventado mediante los vectores de Stokes, y la segunda una forma de manipularlos. Con este objetivo se genera el **cálculo de Mueller** (desarrollado por Hans Mueller en 1943 [2]), con el fin de poder modelar los elementos polarizadores que permiten **modificar el estado de polarización** de la luz. Estos elementos son el polarizador lineal, el retardador y el rotador. El cálculo de Mueller es un método matricial que permite manipular los vectores de Stokes con el fin de aplicar un determinado elemento óptico o polarizador a un estado concreto de polarización. Cada uno de estos elementos es representado por una matriz  $4 \times 4$ , que multiplicar al vector  $4 \times 1$  que conforman los parámetros de Stokes.

$$\begin{pmatrix} I' \\ Q' \\ U' \\ V' \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{pmatrix} \begin{pmatrix} I \\ Q \\ U \\ V \end{pmatrix} \quad (2.7)$$

Esta matriz compuesta por 16 coeficientes nos permite manipular los vectores de Stokes y realizar diversas operaciones que veremos más adelante. Evidentemente, estos coeficientes no pueden tomar cualquier valor, ya que daría lugar a un resultado sin sentido físico. Lo que conseguimos con estas operaciones es el equivalente a pasar el haz de luz por un elemento polarizador, como podría ser un polarizador circular. Esta



representación de los elementos polarizadores tiene la ventaja de permitir representar la acumulación de filtros mediante un producto matricial. Así, aplicar un filtro RCP y después un polarizador lineal a 0 grados es equivalente a multiplicar ambas matrices. Los tipos de elementos necesarios para cambiar todos los parámetros posibles de la elipse de polarización son los siguientes.

### 2.4.1. Polarizador lineal

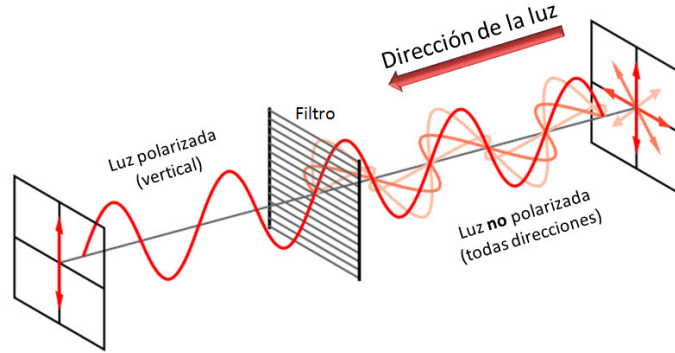


Figura 2.5: Polarizador lineal: la onda oscila solo en una dirección tras atravesar este elemento polarizador.

Un **polarizador lineal** es un atenuador anisotrópico que atenúa las componentes ortogonales de un haz de luz de manera desigual. Las componentes ortogonales del haz incidente están representadas por sus amplitudes  $E_x$  y  $E_y$ , y definimos  $E'_x$  y  $E'_y$  como las amplitudes del haz de salida tras pasar por el polarizador. La relación entre estos dos elementos depende directamente de dos coeficientes de atenuación,  $p_x$  y  $p_y$ , ambos comprendidos entre 0 y 1. Esta relación se define como:

$$E'_x = p_x E_x, \quad E'_y = p_y E_y \quad (2.8)$$

Si en una componente no va a haber atenuación, ese coeficiente valdrá 1, mientras que para la atenuación completa valdrá 0. Dados los coeficientes  $p_x$  y  $p_y$ , la matriz de Mueller de un polarizador lineal es:

$$M(p_x, p_y) = \frac{1}{2} \begin{pmatrix} p_x^2 + p_y^2 & p_x^2 - p_y^2 & 0 & 0 \\ p_x^2 - p_y^2 & p_x^2 + p_y^2 & 0 & 0 \\ 0 & 0 & 2p_x p_y & 0 \\ 0 & 0 & 0 & 2p_x p_y \end{pmatrix} \quad (2.9)$$

Sustituyendo en esta matriz  $p_x = 1$  y  $p_y = 0$  se obtiene el polarizador lineal horizontal ideal. Como se explicará en la sección 2.4.4, este polarizador lineal sirve para generar todos los polarizadores lineales a  $\alpha$  grados al combinar este elemento

con un rotador. Así pues, la siguiente matriz representa un polarizador lineal a  $0^\circ$  (polarización horizontal):

$$M_{0^\circ} = M(1, 0) = \frac{1}{2} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.10)$$

En el anexo B se analiza más en profundidad este elemento, comentando la forma de obtener la matriz así como otras de sus propiedades.

### 2.4.2. Retardador

El **retardador**, también llamado cambiador de fase, es un elemento polarizador cuya función es introducir un **cambio de fase**  $\varphi$  entre las componentes ortogonales del haz de luz entrante.

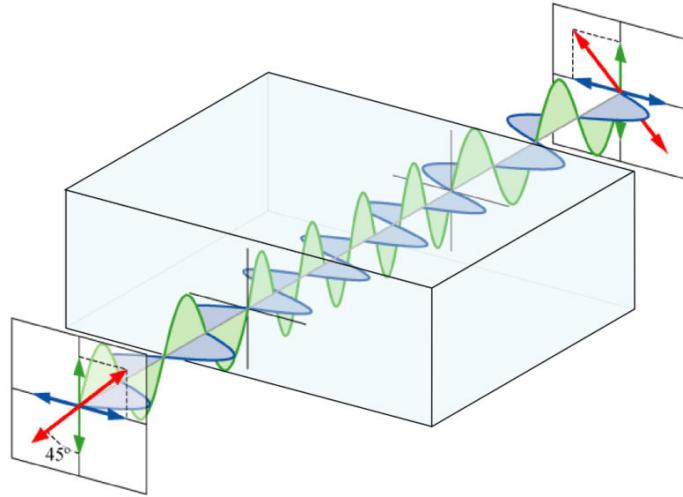


Figura 2.6: Cambio de fase de  $180^\circ$

La matriz de Mueller para un retardador ideal es la siguiente:

$$M(\varphi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos \varphi & \sin \varphi \\ 0 & 0 & -\sin \varphi & \cos \varphi \end{pmatrix} \quad (2.11)$$

Los retardadores más usados son el retardador de un cuarto de onda (QWP por sus siglas en inglés) y el de media onda (HWP), cuyas matrices son las siguientes:

$$M_{QWP,X}(90^\circ) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \quad (2.12)$$

$$M_{HWP,X}(180^\circ) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (2.13)$$

Es especialmente interesante el filtro QWP ya que una de sus propiedades resulta muy útil para la construcción de los vectores de Stokes. El QWP tiene la propiedad de transformar una polarización lineal a  $45^\circ$  en una polarización circular a izquierdas. De la misma forma, se puede utilizar para transformar una polarización lineal a  $135^\circ$  en una polarización circular a derechas. La demostración de este efecto así como otras propiedades del retardador se han incluido en el anexo B

### 2.4.3. Rotador

El **rotador** tiene la función de rotar las componentes ortogonales  $E_x$  y  $E_y$  un ángulo  $\theta$ , dando lugar a  $E'_x$  y  $E'_y$  como se ve en la Figura 2.7.

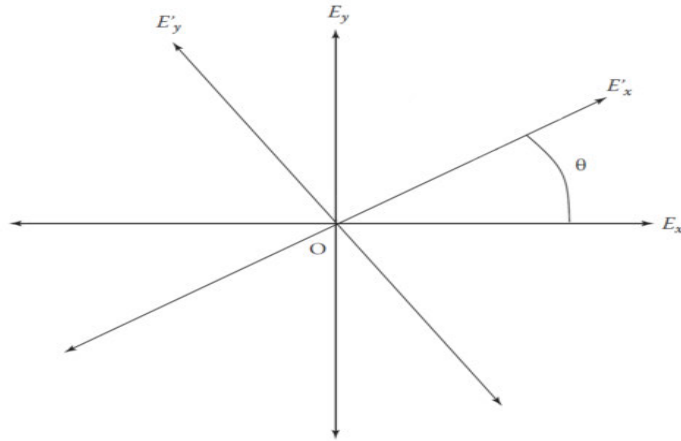


Figura 2.7: Ejes del plano de polarización rotados  $\theta$  grados

Si se proyecta  $(E_x, E_y)$  en los nuevos ejes  $E'_x$  y  $E'_y$ , obtenemos la siguiente relación:

$$E'_x = E_x \cos \theta + E_y \sin \theta, \quad E'_y = -E_x \sin \theta + E_y \cos \theta \quad (2.14)$$

La matriz de Mueller para un rotador de  $\theta$  grados es la siguiente:

$$M_{ROT}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 2\theta & \sin 2\theta & 0 \\ 0 & -\sin 2\theta & \cos 2\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.15)$$

El rotador por si solo no tiene ninguna utilidad. Es cuando se combina con otros elementos cuando tiene utilidad, en especial al utilizarlo junto a un polarizador lineal, ya que nos permite generar cualquier polarizador lineal a  $\theta$  grados.

#### 2.4.4. Combinaciones de elementos

Con los elementos mostrados hasta ahora solo se pueden conseguir un número limitado de filtros (por ejemplo, es imposible generar un polarizador lineal a  $60^\circ$ ). Para ello, así como para generar cualquier polarizador lineal, lo que se hace es **combinar un polarizador lineal con un rotador**.

En la sección del polarizador lineal (2.4.1) se explicaban los filtros de  $0^\circ$  y  $90^\circ$ , ya que los ejes del elemento polarizador están alineados con los ejes del haz. Esto cambia con el rotador, que nos permite rotar estos ejes, evitando que estén alineados con los ejes del haz, dando lugar a todos los filtros lineales distintos de  $0^\circ$  o  $90^\circ$ . La idea es aplicar el polarizador lineal rotando los ejes del mismo, lo que nos permitirá generar un polarizador lineal en ángulos como  $45^\circ$ . Para ello, podemos utilizar el polarizador lineal a  $0^\circ$  y rotarlo  $45^\circ$ . La matriz de Mueller de un polarizador lineal a  $\theta$  grados se obtiene cogiendo la de un polarizador lineal horizontal y rotándolo de la siguiente manera, haciendo uso de las ecuaciones 2.10 y 2.15:

$$M_{PolLineal}(\theta) = M_{ROT}(-\theta)M_{0^\circ}M_{ROT}(\theta) \quad (2.16)$$

Resolviendo obtenemos la matriz final:

$$M_{PolLineal}(\theta) = \frac{1}{2} \begin{pmatrix} 1 & \cos 2\theta & \sin 2\theta & 0 \\ \cos 2\theta & \cos^2 2\theta & \sin 2\theta \cos 2\theta & 0 \\ \sin 2\theta & \sin 2\theta \cos 2\theta & \sin^2 2\theta & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.17)$$

A continuación se muestra la matriz del polarizador lineal a  $45^\circ$  y a  $135^\circ$ :

$$M_{PolLineal}(45^\circ) = \frac{1}{2} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.18)$$

$$M_{PolLineal}(135^\circ) = \frac{1}{2} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.19)$$

# Capítulo 3

## Filtros polarizadores en post-proceso

Como se comentó en la introducción, el objetivo de este TFG es poder aplicar filtros polarizadores sobre imágenes mediante software. Para ello, en el capítulo anterior se han presentado los conocimientos teóricos necesarios para cumplir este objetivo, como son los vectores de Stokes y las matrices de Mueller. Este capítulo se divide en varias secciones, en primer lugar, la captura de imágenes y la generación de las imágenes de Stokes, posteriormente la aplicación de filtros polarizadores sobre imágenes, y por último, la aplicación de filtros para maximizar o minimizar distintos parámetros.

### 3.1. Imágenes de Stokes

En el capítulo 2.3 se han introducido los vectores de Stokes, que permiten representar el estado completo de polarización de un haz de luz. Para poder editar y operar con la polarización de una escena, es necesario generar su imagen de Stokes correspondiente. Para ello, se genera un **nuevo formato de imagen** que consta de **12 canales**. Estos canales se subdividen en 4 grupos de 3 canales cada uno, correspondiéndose cada grupo con un parámetro de Stokes (I, Q, U y V), de forma que para cada parámetro se almacenan sus 3 canales R, G y B. De manera usual, cuando se utiliza la representación RGB, cada uno de estos 3 canales recibe un valor entre 0 y 255. Sin embargo, como se vio en la sección 2.3, los parámetros de Stokes toman valor entre -1 y 1, por lo que los canales de la imagen de Stokes toman estos valores, en lugar del clásico intervalo 0-255. Este almacenamiento ocupa por tanto 4 veces más espacio que una imagen RGB estándar.

Para poder visualizar los parámetros separados, se visualiza cada grupo por separado. El parámetro I siempre tiene un valor positivo entre 0 y 1, por lo que bastaría con multiplicar este valor por 255 para obtener un RGB clásico. Por el contrario, el

resto de parámetros puede oscilar entre -1 y 1. A continuación se muestran la imagen de Stokes para una escena concreta. Como los valores de Q, U y V pueden ser negativos (no se puede mostrar un color negativo), se aplica una normalización para mostrar los parámetros. Esta normalización es tan simple como:

$$Q_{nor} = \frac{Q}{I} + 1 \quad U_{nor} = \frac{U}{I} + 1 \quad V_{nor} = \frac{V}{I} + 1 \quad (3.1)$$



Figura 3.1: Visualización de los parámetros de Stokes de una escena utilizando la normalización presentada en la ecuación 3.1.

Para generar estas imágenes es necesario utilizar una de las definiciones alternativas de los parámetros de Stokes mostradas en 2.3:

$$\begin{aligned} I &= I_{0^\circ} + I_{90^\circ} \\ Q &= I_{0^\circ} - I_{90^\circ} \\ U &= I_{45^\circ} - I_{135^\circ} \\ V &= I_{rcp} - I_{lcp} \end{aligned}$$

Para generar la imagen del parámetro I, se necesitan utilizar dos imágenes, una filtrada a  $0^\circ$  y otra a  $90^\circ$ , al igual que para el parámetro Q. Para el parámetro U son necesarias otras dos imágenes, filtradas a  $45^\circ$  y  $135^\circ$  grados, y por último, para el parámetro V, dos imágenes con un filtro circular, una a derechas y otra a izquierdas. En total son necesarias 6 imágenes para generar el vector de Stokes de una escena. El coste de generar las imágenes de Stokes es lineal en el numero de pixeles, que dependerá de la resolución utilizada.

## 3.2. Captura de imágenes

Como se acaba de comentar, son necesarias **6 imágenes** para poder generar los vectores de Stokes asociados a esa escena. En este trabajo se han utilizado dos conjuntos de imágenes. Por un lado el conjunto que se utilizo en [3](ver Figura 3.2), y otro grupo capturado para este trabajo. En las capturas realizadas para este trabajo no se disponía de un filtro de polarización circular, por lo que el parámetro V no se puede generar. Sin embargo, este parámetro solo interviene en las secciones 3.3.2 y 3.3.3,

para el polarizador circular y el polarizador por parámetros, ya que los polarizadores lineales no hacen uso del parámetro V. Para tomar las imágenes se ha utilizado una cámara reflex Canon eox 20D, con un portafiltros acoplado y un polarizador lineal. Para todas las capturas se ha utilizado un trípode, ya que es necesario mantener el mismo encuadre de la escena en todas las capturas. Además del trípode, se ha utilizado un disparador remoto para minimizar los cambios de encuadre que pueda sufrir la cámara entre captura y captura. En la Figura 3.3 se pueden observar las distintas imágenes tomadas con este método.



Figura 3.2: Capturas de 2 escenas distintas capturadas mediante el método presentado en [3]

Sobre el portafiltros se han realizado 4 muescas distintas, cada una marcando los grados de cada captura necesaria ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$ ). Con las muescas realizadas, se toman las 4 imágenes siguiendo siempre el mismo orden para posteriormente poder identificarlas. Este método, aunque funcional y barato, tiene el contra de la precisión y la consistencia. En primer lugar no se pueden tomar todas las capturas a la vez, lo que acarrea varios problemas. Además, la precisión a la hora de colocar el filtro al ángulo correcto depende del ojo humano, por lo que en general no tendremos tomas perfectas de cada ángulo, sino que habrá un pequeño desfase, lo que resulta en parámetros de Stokes no exactos.



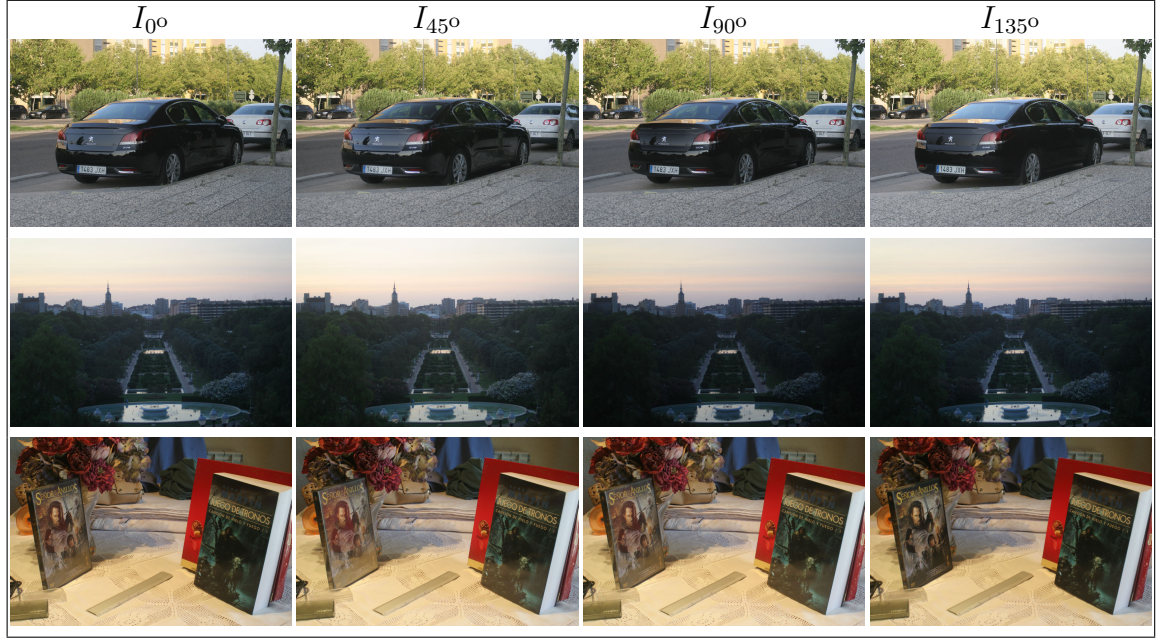


Figura 3.3: Capturas de 3 escenas distintas capturadas mediante el método presentado en la Sección 3.2

Uno de los principales problemas de este método de captura es que son necesarias dos fotografías para generar cada parámetro. Al tomar cada una de las imágenes de forma separada en distintos instantes, es posible que entre ambas imágenes haya cambiado la escena, por ejemplo, al fotografiar un objeto móvil como un coche. Esto entre otras cosas puede generar que un objeto aparezca en una toma, la de  $0^\circ$  por ejemplo, y no aparezca en la de  $90^\circ$ . Esto generará una apariencia translúcida del objeto, dando lugar a posteriores resultados erróneos. Un ejemplo de este problema se puede observar en la Figura 3.4.



Figura 3.4: Parámetro I de una imagen de Stokes, generada a partir de dos imágenes en las que un objeto aparece en una captura y no en la otra.

Además del problema anterior, existen otros que pueden evitar alcanzar el objetivo deseado, por ejemplo, eliminar los reflejos en el metal de un coche(ver Sección 3.4.2).



Esto ocurre porque el reflejo que se quiere eliminar no esta perfectamente alineado en las 6 imágenes que intervienen al generar el vector. Esto conlleva que al intentar eliminar el reflejo, este no se pueda eliminar del todo ya que dicho reflejo estará en posiciones distintas en cada imagen. Lo mismo podría ocurrir si la fuente de luz ha cambiado de intensidad o de posición.



(a) Original (b) Reflejo bien minimizado (c) Reflejos mal eliminados

Figura 3.5: Ejemplo del problema de alineación

En conclusión este **método** cuenta con la ventaja de ser **barato, simple y fácilmente reproducible** si se compara con métodos más avanzados, permitiendo pese a los problemas comentados, capturar distintos tipos de escena y obtener buenos resultados.

Para este problema existen varias investigaciones de cámaras capaces de capturar los parámetros de Stokes en una sola captura. Algunas de estas propuestas consisten en micro filtros polarizadores alineados con filtros de Bayer para generar todos los parámetros en una captura [7].

### 3.3. Filtrado software

En esta sección se van a tratar la aplicación de **filtros polarizadores en fase de postproceso**. Esto permite aplicar cualquier tipo de filtro que se pudiera aplicar en fotografía, pero mediante software de forma casi inmediata. Para ello se van a utilizar las matrices de Mueller explicadas en la sección 2.4.

Si recordamos el cálculo de Mueller, en el se multiplica el vector de Stokes,  $S$ , por una matriz,  $M$ , para obtener un nuevo vector filtrado  $S'$ . En este caso, solo se desea obtener la imagen resultante de aplicar el filtro, por lo que no es necesario calcular todo el vector  $S'$ , sino que basta con calcular el parámetro  $I'$ . De esta forma, el algoritmo general a seguir sera calcular  $I'$ , multiplicando el vector  $S$  por la primera fila de la matriz  $M$ . Como es necesario calcular la imagen  $I'$ , cada pixel de  $I'$  sera el resultado de multiplicar el pixel correspondiente de  $I$ ,  $Q$ ,  $U$  y  $V$  por la primera fila de  $M$ .

$$I'(i, j) = m_{11}I(i, j) + m_{12}Q(i, j) + m_{13}U(i, j) + m_{14}V(i, j) \quad (3.2)$$

El coste de este tipo de algoritmos se puede calcular de forma muy sencilla, ya que será el coste de realizar la operación anterior a cada píxel de la imagen, es decir, en una imagen de  $n \times m$  píxeles tendrá coste cuadrático  $O(n * m)$

### 3.3.1. Filtro de polarización lineal

En la sección de Mueller 2.4.1 se ha visto la matriz correspondiente para la polarización lineal dado un ángulo  $\alpha$ . Si extraemos la primera fila de la misma, obtenemos que la intensidad de un píxel  $I'(i, j)$  filtrado  $\alpha$  grados será:

$$I'(i, j)(\alpha) = \frac{1}{2}(I(i, j) + Q(i, j) \cos 2\alpha + U(i, j) \sin 2\alpha) \quad (3.3)$$



Figura 3.6: Imagen filtrada con un polarizador lineal a  $75^\circ$

Aplicando esto a todos los píxeles de la imagen, obtenemos una imagen filtrada a  $\alpha$  grados. La imagen filtrada es una combinación lineal de las imágenes I, Q y U, con lo que si solo se quisieran realizar filtros lineales, serían **necesarias 4 imágenes en lugar de 6**. Se puede observar un polarizador lineal a  $75^\circ$  en la Figura 3.6.

### 3.3.2. Filtro circular

Como se ha visto en la sección de las matrices de Mueller (ecuación B.15), para generar un filtro circular se utiliza un retardador de un cuarto, QWP, y un filtro lineal a  $45^\circ$ . Si utilizamos un filtro QWP con el eje rápido en X, obtendremos un filtrado circular a izquierdas, mientras que si se utiliza un QWP siendo Y el eje rápido, obtenemos una polarización circular a derechas:

$$S' = M_{QWP,X} S_{45^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix} \quad (3.4)$$

$$S' = M_{QWP,Y} S_{45^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (3.5)$$

De esta forma, las matrices de Mueller para un RCP y un LCP serán de la siguiente forma:

$$M_{RCP} = M_{QWP,Y} M_{45^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \frac{1}{2} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad (3.6)$$

$$M_{LCP} = M_{QWP,X} M_{45^\circ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \frac{1}{2} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 \end{pmatrix} \quad (3.7)$$

Una de las propiedades del polarizador circular es que por un lado se comporta como un polarizador lineal y por otro como un polarizador circular. La parte del polarizador lineal depende del ángulo al que se aplique el polarizado lineal, siendo la intensidad de esta parte  $I_l(\alpha)$ :

$$I_l(\alpha) = \frac{1}{2}(I + Q \sin(\alpha) + U \cos(\alpha)) \quad (3.8)$$

Por otro lado, la parte circular no se ve afectada por la rotación, ya que la intensidad de salida del filtro circular sera:

$$I_c = \frac{1}{2}(I + V) \quad (3.9)$$

Se puede observar el resultado de aplicar un polarizador circular a derechas(RCP) en la Figura 3.7.



(a) Original

(b) Filtrada

Figura 3.7: Imagen filtrada con un polarizador circular a derechas

### 3.3.3. Filtro parametrizado

El cálculo de Mueller es al fin y al cabo multiplicar cada componente del vector de Stokes por un valor entre -1 y 1. Una de las principales ventajas de filtrar la imagen

por software es, que al tener el estado completo de polarización de la luz, podemos manipularlo de cualquier forma, y en este caso, generar **filtros que no se podrían diseñar en la realidad**. De esta forma, podemos parametrizar la función de intensidad con 3 valores de la siguiente forma:

$$I(c_1, c_2, c_3) = \frac{1}{2}(I + c_1Q + c_2U + c_3V), \quad -1 \leq c_1, c_2, c_3 \leq 1 \quad (3.10)$$



Figura 3.8: Imagen filtrada con un polarizador por parámetros con  $c_1=1$ ,  $c_2=0$  y  $c_3=-1$

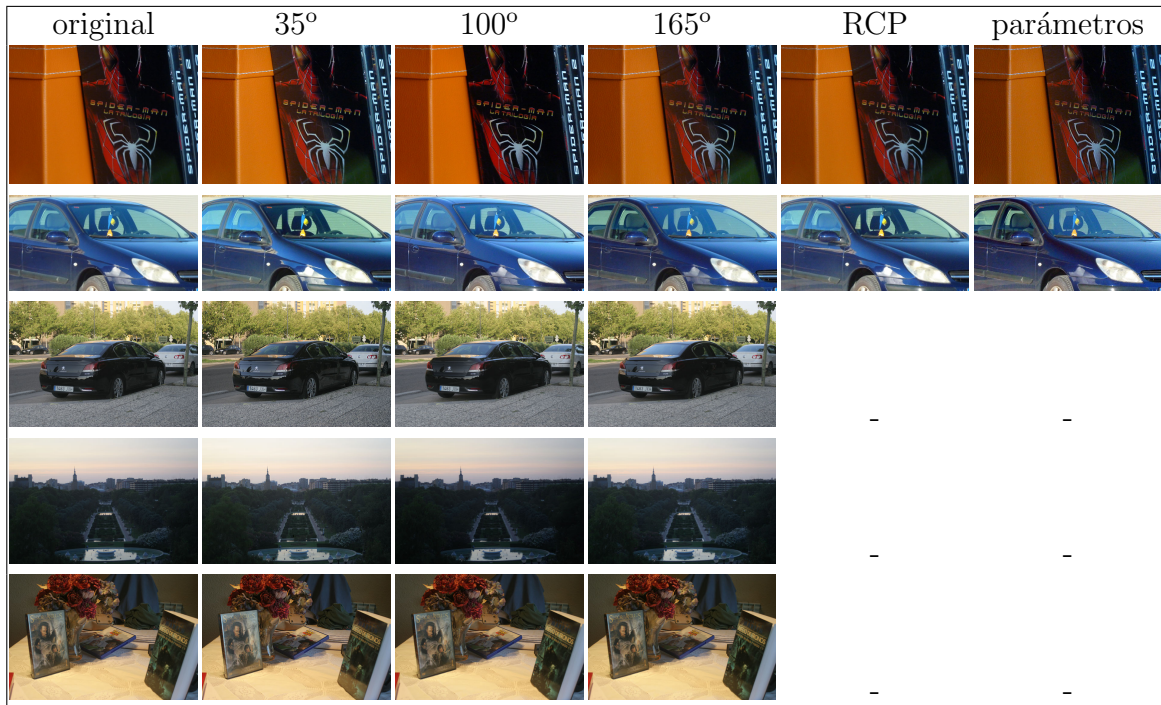


Figura 3.9: Resultados al aplicar distintos filtros(El filtrado por parámetros corresponde a los valores  $c_1=1$ ,  $c_2=-1$ ,  $c_3=-1$ )

En la Figura 3.9 se pueden observar distintos resultados de aplicar varios filtros. A las últimas tres escenas no se les pueden aplicar filtros circulares ni filtros por



parámetro, debido a que no se dispone de las imágenes correspondientes al filtro circular, por lo que no se puede generar el parámetro V. Sobre los tres filtros lineales, hay que destacar que no se han tomado imágenes con esos filtros (30°, 100° y 165°), sino que se han generado a partir de las 4 imágenes originales. La elección de estos tres ángulos es arbitraria, ya que se podrían generar resultados para cualquier ángulo de un polarizador lineal. Además reseñar que con el filtrado por parámetros se pueden generar filtros que no existen en la realidad.

### 3.4. Filtros de optimización

En fotografía se utilizan filtros de polarización con varios objetivos, por ejemplo, maximizar o minimizar el contraste o el brillo en una imagen. Esta misma idea se puede aplicar al filtrado software, de forma que de manera automática, encuentre el filtro óptimo para minimizar el brillo de una imagen, por ejemplo (y minimizar los reflejos). En el caso de un fotógrafo, este rotaría el filtro lineal hasta que obtuviera el ángulo que minimiza dicho brillo, pero si aplicamos esta idea a un software podemos encontrar el ángulo óptimo exacto, minimizando en este caso la función de brillo. El problema se reduce a encontrar el **ángulo** con el que un **polarizador lineal maximiza o minimiza el parámetro** deseado. Esta misma idea ya se había aplicado en trabajos anteriores [3], utilizando el mismo principio que el fotógrafo en sus capturas, es decir, probando en un número elevado de ángulos y seleccionando aquel que mejor resultado da para el parámetro seleccionado. En este TFG, el procedimiento para calcular dichos ángulos óptimos es distinto, y se basa en el cálculo de la derivada de la función que define el parámetro a optimizar. Al realizar un cálculo analítico, no es necesario depender de la prueba y error, lo que da lugar a **algoritmos más rápidos** y capaces de obtener el **resultado exacto** para un parámetro dado. Como se verá en el capítulo siguiente, esto resultará en la posibilidad de aplicar estos filtros para aplicaciones en tiempo real.

Para alcanzar el objetivo de optimización, el procedimiento es muy similar independientemente del parámetro que se quiera optimizar. Al fin y al cabo se desea encontrar los máximos y mínimos de una función  $f(\alpha)$  que define el comportamiento del parámetro a optimizar en función del ángulo de un polarizador lineal. Si se define  $f'(\alpha)$  y  $f''(\alpha)$  como la primera y la segunda derivada de  $f$  respecto a  $\alpha$ , se puede obtener el valor del ángulo que maximiza y minimiza el parámetro:

$$f'(\alpha) = 0, \quad f''(\alpha) \begin{cases} > 0 \rightarrow \text{Minimo} \\ < 0 \rightarrow \text{Maximo} \end{cases} \quad (3.11)$$

De esta forma se pueden obtener los valores exactos en los que dicho parámetro alcanza su valor máximo, o mínimo, al contrario que un algoritmo de prueba y error. Al igual que se realiza esta optimización con un polarizador lineal, se podría realizar con otro tipo de filtros.

### 3.4.1. Filtros globales

Estos algoritmos consisten en dos fases claramente diferenciadas. En primer lugar, calcular el ángulo óptimo para la maximización o minimización del parámetro, y posteriormente su aplicación. Este tipo de filtros realizan una función similar a la que haría un fotógrafo girando el filtro hasta obtener el resultado deseado, pero sin depender de la prueba y error y encontrando el **resultado analíticamente**. Los tiempos obtenidos son cercanos a 1 segundo, siendo el más rápido el de la luminosidad (100ms), debido a que los otros dos parámetros, el contraste y la saturación, requieren resolver una ecuación de grado 4. Estos tiempos se han obtenido sobre imágenes de 3504x2336 píxeles.

#### Optimización de la luminosidad

La **luminosidad** es un indicador sobre el aspecto que tiene un color. Cuanto más oscuro es el color, su luminosidad es más débil. Este término se asocia normalmente con brillo, luminancia, etc. La luminosidad de un color queda definida entre dos colores que marcan los extremos, siendo el negro el color menos luminoso y el blanco el más luminoso.

La luminosidad es especialmente interesante por varias razones. La más importante tiene que ver con las propiedades de algunos materiales comunes, como los metales o los cristales. Estos materiales polarizan la luz cuando esta interactúa con ellos, ya sea reflejándola o refractándola. Esta propiedad permite que al tomar fotografías de materiales de este tipo, sea posible **eliminar** total o parcialmente los **reflejos** y brillos de estas superficies. En este caso, cuando aparezcan dichos reflejos aumentaría la luminosidad de la imagen, y al contrario cuando estos no aparezcan.

El primer paso es definir la función que define la luminosidad,  $L$ . En un espacio de color RGB,  $L = p_R * R + p_G * G + p_B * B$ , donde  $p_x$  es el valor con el que se pondera cada canal. La suma de estos tres pesos tiene que ser 1. El ojo humano es más sensible al color verde que al resto a la hora de percibir la intensidad de la luz, por lo que esto

se corrige ponderando cada color de forma distinta [8]. La luminosidad de un pixel  $p$  es:

$$L(p) = 0,2126R(p) + 0,7152G(p) + 0,0722B(p) \quad (3.12)$$

Posteriormente, se calcula la función de la luminosidad de un polarizador lineal rotado  $\alpha$  grados. Para ello, hacemos uso de la ecuación 3.3 que se definía para aplicar un filtro lineal, y se combina con la función de luminosidad. La luminosidad de un pixel filtrado  $\alpha$  grados es:

$$\begin{aligned} I(p) &= k_1 I_R(p) + k_2 I_G(p) + k_3 I_B(p), & Q(p) &= k_1 Q_R(p) + k_2 Q_G(p) + k_3 Q_B(p) \\ U(p) &= k_1 U_R(p) + k_2 U_G(p) + k_3 U_B(p) \end{aligned}$$

- $p$  representa la posición del pixel.
- $I_R(p), I_G(p), I_B(p)$  representan los tres canales RGB del parámetro  $I$  en el pixel  $p$ .
- $Q_R(p), Q_G(p), Q_B(p)$  representan los tres canales RGB del parámetro  $Q$  en el pixel  $p$ .
- $U_R(p), U_G(p), U_B(p)$  representan los tres canales RGB del parámetro  $U$  en el pixel  $p$ .
- $k_1, k_2, k_3$  representan los pesos de cada canal. Según la ecuación 3.12, estos valores serán, 0.2126, 0.7152 y 0.0722 respectivamente.

$$L(\alpha, p) = \frac{1}{2}(I(p) + Q(p) \cos 2\alpha + U(p) \sin 2\alpha) \quad (3.13)$$

Con la luminosidad de un pixel definida, la luminosidad global se define como el sumatorio de la luminosidad de cada pixel:

$$L(\alpha) = \frac{1}{n} \sum_{p \in P} L(\alpha, p) \quad (3.14)$$

- $P$  es el conjunto de pixeles de la imagen.
- $n$  es el número de pixeles.

Obteniendo la **derivada** de esta función y calculando sus **puntos críticos**, donde  $L'(\alpha) = 0$ , se obtienen los valores de  $\alpha$  que maximizan o minimizan el brillo. Si calculamos también la **segunda derivada** de la función, podremos conocer si el punto crítico es un máximo o un mínimo en función del valor de  $L''(\alpha)$ .

$$Q = \sum_{p \in P} Q(p) \quad U = \sum_{p \in P} U(p)$$

$$L'(\alpha) = U \cos 2\alpha - Q \sin 2\alpha \quad (3.15)$$

$$L''(\alpha) = -2Q \cos 2\alpha - 2U \sin 2\alpha \quad (3.16)$$

Para obtener el punto crítico,  $L'(\alpha) = 0$ :

$$L'(\alpha) = 0 \Rightarrow \tan 2\alpha = \frac{U}{Q} \quad (3.17)$$

Con esto obtendremos dos valores posibles de  $\alpha$ :

$$\alpha = \frac{\arctan\left(\frac{U}{Q}\right)}{2}$$

$$\alpha = \frac{\arctan\left(\frac{U}{Q}\right) + \pi}{2}$$

Esta segunda raíz se debe a que la función  $\tan(\alpha)$  es periódica en  $\pi$ . Después, evaluando  $L''(\alpha)$  con estos dos valores podemos conocer si se trata de un máximo o un mínimo.

Con ambos valores de  $\alpha$  calculados, se comprueba cuál es el máximo y cuál el mínimo, y se aplica un filtrado lineal con dicho ángulo a la imagen. Se puede ver el resultado en la siguiente Figura 3.10.



(a) Original



(b) Maximización



(c) Minimización

Figura 3.10: Resultado de aplicar el filtro de optimización del brillo global. En la maximización se ha aplicado un filtro lineal a 130.22 grados, mientras que en la minimización uno a 40.22 grados.

En la Figura 3.10, se puede observar el filtro global del brillo sobre una escena con un coche. Se puede ver como en el techo y la luneta de ambos coches, las zonas donde aparecen los reflejos de más intensidad, se intensifican o minimizan. Por ejemplo en la luneta, en el caso de la maximización se ve un reflejo muy fuerte, mientras que en la minimización se puede ver el interior del coche.

### Optimización del contraste

Al igual que se ha realizado con la luminosidad, se va a aplicar la misma idea para maximizar o minimizar el contraste de una imagen. El **contraste**, al contrario que el



caso del brillo, no depende únicamente de un pixel, sino que el valor del contraste de un pixel depende de los que tiene alrededor. El contraste se define como la diferencia de color que hace un objeto distinguible del resto de la imagen. Hay varias definiciones del contraste, en algunas se hace referencia al color, mientras que otras solo trabajan con la luminosidad. Se ha utilizado esta última, definiendo el contraste de una ventana de pixeles como:

$$C(W) = \sqrt{\frac{1}{|W|} \sum_{i \in W} \left( L(i) - \overline{L(W)} \right)^2} \quad (3.18)$$

- $W$  es la ventana de pixeles.
- $L(i)$  es la luminosidad de un pixel.
- $\overline{L(W)}$  es la media de la luminosidad en la ventana.

El procedimiento a seguir es el mismo que para la optimización de la luminosidad. En primer lugar se sustituye en la definición del contraste  $C$  las ecuaciones de un filtro lineal, se deriva esa función  $C(\alpha)$  y se obtienen sus raíces.

$$C'(W, \alpha) = \frac{\frac{1}{|W|} \sum_{i \in W} 2 \left( L(i, \alpha) - \overline{L(W, \alpha)} \right) \left( L'(i, \alpha) - \overline{L'(W, \alpha)} \right)}{2 \sqrt{\frac{1}{|W|} \sum_{i \in W} \left( L(i, \alpha) - \overline{L(W, \alpha)} \right)^2}} \quad (3.19)$$

Estos cálculos se adjuntan en el anexo C.

Para obtener el contraste global, definimos el contraste global como el contraste, definido en la ecuación 3.18, cuando la ventana  $W$  es toda la imagen. Al final, para extraer las raíces hace falta resolver una **ecuación de grado cuatro**, lo cual acarrea varios problemas. El primero es que resolver este tipo de ecuaciones mediante el método general (el equivalente de  $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$  para ecuaciones de grado 2) es muy costoso para utilizarse en la mayoría de casos, ya que requiere un número elevadísimo de operaciones.

Hay diferentes métodos para resolver este tipo de ecuaciones. En este caso se ha utilizado una librería de C++ que utiliza un método simplificado [9]. Al resolver esta ecuación, llegamos a tres casos diferentes:

- Se obtienen 4 raíces reales, sobre las que se buscará si se tratan de mínimos o máximos locales o globales.
- Se obtienen 2 raíces reales y 2 complejas, buscando sobre las reales. En el caso de que ninguna de estas raíces fuera un máximo, en el caso de la maximización, o un mínimo, para la minimización, se aplicaría el mismo caso que para 4 raíces complejas.

- Se obtienen 4 raíces complejas, en cuyo caso, no se puede obtener el ángulo óptimo para minimizar o maximizar el contraste, por lo que se busca el que maximice (o minimice) el contraste de un subconjunto de ángulos predeterminados entre  $0^\circ$  y  $180^\circ$ . Se divide este intervalo en  $n$  partes iguales y se comprueba cual de estos  $n$  ángulos consigue mejor el objetivo. Si se escoge un  $n$  muy grande, la ejecución del algoritmo será muy lenta. Probando con distintos valores de  $n$ , se ha seleccionado un valor de  $n$  entre 7 y 14 .

Los resultados obtenidos son los siguientes:



Figura 3.11: Resultados de aplicar los filtros de optimización globales del contraste. Se puede observar el principal cambio en el cielo.

En la Figura 4.6 se pueden observar los resultados obtenidos con el filtro global del contraste. En ambas imágenes, la principal diferencia se puede ver en el cielo. El índice de contraste entre ambas imágenes se corresponde con el resultado, siendo menor el de la minimización y mayor el de la maximización. Sin embargo, el contraste puede parecer mayor en el caso de la minimización, especialmente en las imágenes de la primera fila. Esta diferencia entre la apreciación y el resultado se debe a que en la función del contraste solo se utiliza la luminosidad de cada color.

### Optimización de la saturación

La **saturación** se define como la intensidad de un matiz específico en un color. Un color muy saturado tiene un color intenso, mientras que uno menos saturado, tiende al gris. Matemáticamente, y en concreto en el modelo de color RGB, se define la saturación

de un color, siendo  $\mu = (R + G + B)/3$ , como:

$$S(p) = \sqrt{\frac{(R - \mu)^2 + (G - \mu)^2 + (B - \mu)^2}{3}} \quad (3.20)$$

Al igual que en los dos apartados anteriores, es necesario calcular  $S'(p, \alpha)$  para obtener los puntos críticos de  $S(p, \alpha)$ . El cálculo de esta derivada, así como sus raíces se adjunta en el anexo C.

Al igual que sucedía en la optimización del contraste, para obtener las raíces de esta función es necesario resolver un polinomio de grado cuatro. Se aplica el mismo algoritmo que para el contraste, si hay 4 o 2 raíces reales, se busca sobre esas raíces cual de los ángulos maximiza (o minimiza) la saturación. En el caso que no existan raíces reales, se busca sobre un número predeterminado de ángulos cual es el mejor ángulo.

En cuanto a los resultados, los cambios en la saturación que se pueden conseguir mediante filtros polarizadores son limitados. En primer lugar dependen mucho de la escena que se ha capturado, así como la luz que incide sobre los cuerpos que aparecen. Los mejores resultados se pueden observar en los cielos claros, como se podía ver en la figura de la introducción 1.2, mientras que en otras imágenes las diferencias son mucho mas sutiles.



Figura 3.12: Resultados de aplicar los filtros de optimización globales de la saturación. Se pueden observar los cambios en el cielo y en la vegetación.

Aunque el cambio es sutil se puede observar en detalles como el verde de los árboles la diferencia de color. También se observa el problema que se comentaba en la sección

3.2, ya que en las distintas tomas se puede ver que los árboles están un poco borrosos, debido a que a la hora de tomar las fotografías para generar la imagen de Stokes, hay movimiento entre distintas capturas debido al viento. En especial, se puede apreciar como el cielo de la segunda fila de imágenes cambia en los dos casos. En la maximización se ve un azul más vivo, mientras que en la minimización el azul del cielo esta más aclarado hacia el blanco.

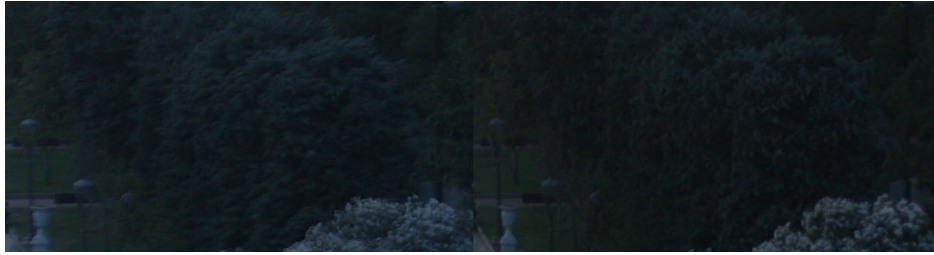


Figura 3.13: Detalle de una imagen de la Figura 4.6 en la optimización de la saturación donde se observa un cambio sutil en el color de los arboles

### 3.4.2. Filtros locales

Al tratar la imagen mediante software, **cada pixel** puede ser **tratado de forma independiente** al resto de pixeles de la imagen, cosa que desde la fotografía convencional es imposible de conseguir, y que se puede explotar para conseguir **resultados imposibles mediante la fotografía**. Estos filtros aplican un filtro lineal con distinto ángulo a cada pixel de la imagen, siendo este ángulo el que maximiza (o minimiza) el parámetro en ese pixel. A estos filtros los llamaremos filtros locales. Al contrario que en los filtros globales, que se calculaba un único ángulo y se aplicaba a toda la imagen, ahora se calcula un ángulo por cada pixel de la imagen. Esto incrementa el coste, especialmente en el caso del contraste, mientras que en el brillo se obtienen tiempos de 300-400 milisegundos y de poco más de 1 segundo para la saturación.

#### Optimización de la luminosidad

En la sección correspondiente de los filtros globales, se definía la función de luminosidad, así como se calculaba su derivada en la ecuación 3.13. Para el filtro local, evidentemente, la definición de la luminosidad es la misma, así como su derivada. El cambio introducido es, que en lugar de calcular un  $\alpha$  para toda la imagen, se calcula un  $\alpha$  distinto para cada pixel. Este algoritmo lógicamente requiere más tiempo, ya que en lugar de calcular un  $\alpha$ , se realiza ese cálculo para cada pixel de la imagen. Si se utiliza la ecuación 3.13 y se deriva:

$$L'(p, \alpha) = U(p) \cos 2\alpha - Q(p) \sin 2\alpha \quad (3.21)$$

Con lo que las raíces por pixel serán:

$$\alpha = \frac{\arctan\left(\frac{U(p)}{Q(p)}\right)}{2}$$

$$\alpha = \frac{\arctan\left(\frac{U(p)}{Q(p)}\right) + \pi}{2}$$

Con este filtro se obtienen resultados imposibles de conseguir mediante la fotografía convencional. Como se comentaba al inicio de la sección 3.4.1, algunos materiales polarizan la luz al interactuar con ella. Aquí es donde cobran especial interés los filtros locales o por pixel. En el caso de las superficies metálicas, como las que se observaban en la Figura 3.10, cada superficie metálica refleja la luz polarizándola en un ángulo concreto. Al aplicar el mismo filtro a toda la imagen, es imposible minimizar o maximizar los brillos que se reflejan en estas superficies metálicas, ya que cada una refleja la luz con un ángulo de polarización lineal distinta. Con el filtrado por pixel, este problema se elimina, ya que cada pixel tendrá un filtro distinto para minimizar o maximizar esa zona concreta. De esta forma, el resultado a la hora de minimizar o maximizar la luminosidad es mucho mejor:

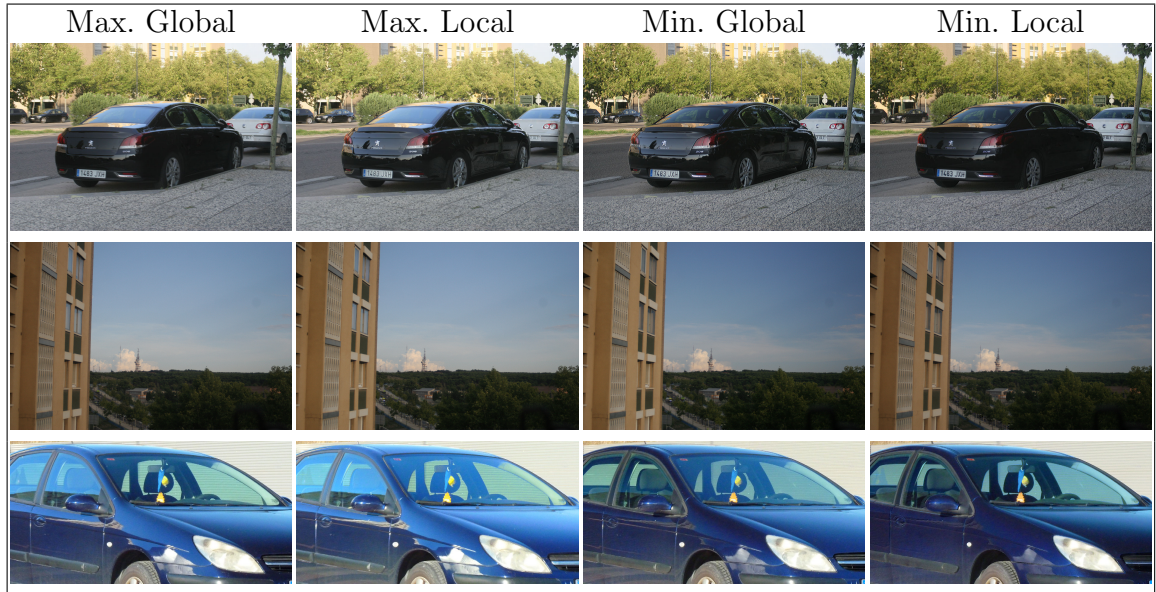


Figura 3.14: Comparativa para distintas escenas entre los filtros de optimización del brillo globales y locales

En algunos casos se consiguen **eliminar completamente los reflejos**, como en el lateral del coche del coche de la primera fila en la Figura 3.14, mientras que otros casos que reciben mucha iluminación tan solo se consiguen minimizar. La comparativa muestra como los filtros globales solo permiten minimizar (o maximizar) la zona de la escena que más aporta a la luminosidad, por el contrario, con el filtro local se elimina



esta limitación, permitiendo minimizar todas las superficies de la imagen. Para observar que ángulo se ha utilizado en cada pixel, se ha utilizado el modelo de color HSV, que se compone de tres componentes: el matiz, la saturación y el valor(luminosidad). Se genera la imagen manteniendo la saturación y el valor al máximo, y el valor del matiz es el valor del ángulo seleccionado. En estas imágenes se puede observar como a lo largo de una superficie curva, como el maletero, el ángulo óptimo va cambiando gradualmente a lo largo de la superficie (ver Figura 3.15).

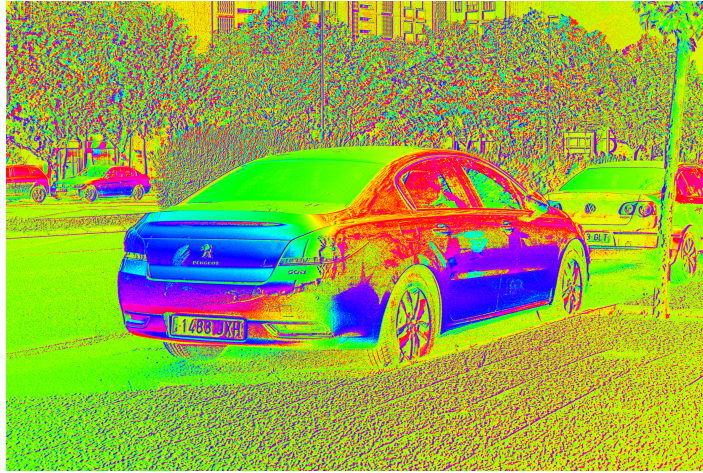
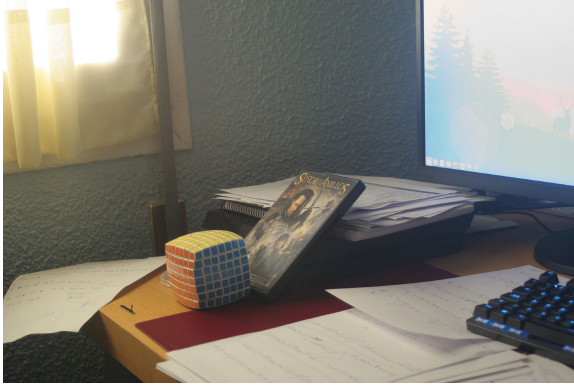
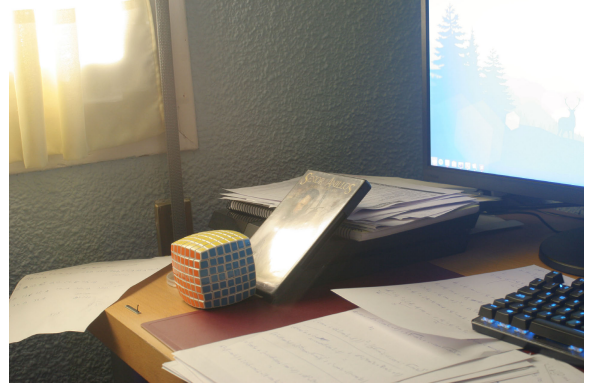


Figura 3.15: Ángulo seleccionado para cada pixel en la maximización local de la luminosidad

Otro ejemplo en el que se puede observar las ventajas de los filtros por pixel frente a los globales, se da cuando en la escena hay alguna fuente de luz polarizada. En la escena que se va a utilizar de ejemplo, aparece una carátula de plástico, que refleja la luz, y una pantalla de ordenador. La luz de las pantallas de ordenador esta polarizada, por lo que si tratamos de utilizar los filtros globales como los filtros por pixel, se puede observar que los globales solo se centran en maximizar o minimizar el brillo de la pantalla (ya que ese brillo aporta más a la imagen que el de la carátula). Por otro lado, con los filtros por pixel, se pueden minimizar ambas superficies.



(a) Maximización global



(b) Maximización local



(c) Minimización global



(d) Minimización local

Figura 3.16: Comparativa entre la optimización global y local de la luminosidad en una escena con una fuente de luz polarizada.

En este ejemplo se pudo observar muy bien el compromiso de los filtros globales. En la escena capturada, queda claro que la superficie que más cambio de luminosidad aporta es la pantalla de ordenador. Esta superficie domina la luminosidad de la escena, por lo que el filtrado global se centra en minimizar esta superficie. Se observan las limitaciones de este tipo de filtros al ver la carátula, que aunque en el filtrado se este minimizando la luminosidad, el brillo de esta no se elimina. Sin embargo, con el **filtrado por pixel** se pueden minimizar o maximizar ambos reflejos, consiguiendo así **objetivos** mucho más **óptimos** que los que se pueden conseguir mediante filtros globales, siendo además irreproducibles mediante la fotografía convencional. Además de esta búsqueda del ángulo óptimo para maximizar o minimizar el brillo utilizando un filtro lineal, también se ha realizado esta búsqueda utilizando el filtro parametrizado (ver Sección 3.3.3). La ecuación de la luminosidad quedaría:

$$L(c_1, c_2, c_3) = \frac{1}{2}(I + c_1Q + c_2U + c_3V) \quad (3.22)$$

En este caso, se sabe que cada coeficiente tiene que estar entre -1 y 1, con lo que obtener el máximo para estos valores es trivial. Se diferencian dos casos, maximizar y

minimizar:

- Maximizar: se quiere obtener el valor más alto de  $L(c_1, c_2, c_3)$ . Por ello el objetivo es que  $c_1Q$  sea un valor positivo, así como  $c_2U$  y  $c_3V$ . De esta forma, se asigna  $c_1 = 1$  si  $Q$  tiene un valor positivo, y  $c_1 = -1$  si el valor de  $Q$  es negativo, de forma que  $c_1Q$  sea siempre un valor positivo para maximizar  $L$ . Esto mismo se realiza con  $c_2U$  y  $c_3V$ .
- Minimizar: se quiere obtener el valor mínimo de  $L(c_1, c_2, c_3)$ . Por ello el objetivo es que  $c_1Q$  sea un valor negativo, así como  $c_2U$  y  $c_3V$ . De esta forma, se asigna  $c_1 = -1$  si  $Q$  tiene un valor positivo, y  $c_1 = 1$  si el valor de  $Q$  es negativo, de forma que  $c_1Q$  sea siempre un valor negativo, haciendo que  $L$  decrezca. Esto mismo se realiza con  $c_2U$  y  $c_3V$ .

## Optimización del contraste

Para la optimización del contraste local la única diferencia en el cálculo es que, en lugar de aplicar una ventana del tamaño de la imagen, se aplica otro tipo de ventana más pequeña. A cada pixel se le aplicará un filtro distinto en función de la ventana asociada a cada pixel. Estas ventanas serán una **ventana gaussiana**, que pondera cada pixel de la ventana con más valor cuanto más cerca este al centro de la ventana. Dicho centro se corresponderá con el pixel que se esta evaluando. La ventana gaussiana a utilizar estará paremetrizada por el tamaño de la ventana y la desviación típica. De esta forma cada pixel tendrá un peso distinto dependiendo de su cercanía al centro de la ventana(el pixel a evaluar)

$$f(d, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d^2}{2\sigma^2}} \quad (3.23)$$

En la figura 3.17 se pueden observar los pesos que se asignarían a una ventana gaussiana 5x5. De esta forma que el pixel más lejano al centro ponderaría 0.17, mientras que un pixel colindante 0.8.

0.17	0.32	0.41	0.32	0.17
0.33	0.64	0.8	0.64	0.33
0.41	0.8	1	0.8	0.41
0.33	0.64	0.8	0.64	0.33
0.17	0.32	0.41	0.32	0.17

Figura 3.17: Valores de una ventana gaussiana 5x5 con  $\sigma = 1,5$

Si recordamos la definición del contraste:

$$C(W) = \sqrt{\frac{1}{|W|} \sum_{i \in W} \left( L(i) - \overline{L(W)} \right)^2} \quad (3.24)$$



Ahora, la normalización del tamaño de la ventana ( $\frac{1}{|W|}$ ), en lugar de hacerse por el número de píxeles de la ventana, se hace por la suma de todos los pesos de la misma. El otro cambio es que en lugar de ser un sumatorio normal, la suma de cada píxel  $i$  esta ponderada por su peso.

El tamaño va a ser clave en el resultado y va a tener un impacto muy importante en el rendimiento del programa. En primer lugar, cuanto más grande sea la ventana, más tiempo de ejecución tendrá el algoritmo. Además el **tamaño de la ventana** tendrá un **impacto muy importante en el resultado**, ya que cuanto más grande sea, el filtrado de cada píxel dependerá de píxeles cada vez más lejanos. A continuación se van a mostrar los resultados obtenidos en la minimización del contraste para distintos tamaños de ventanaradio = 2, 5, 10, 25), y la imagen con el ángulo seleccionado para cada píxel:

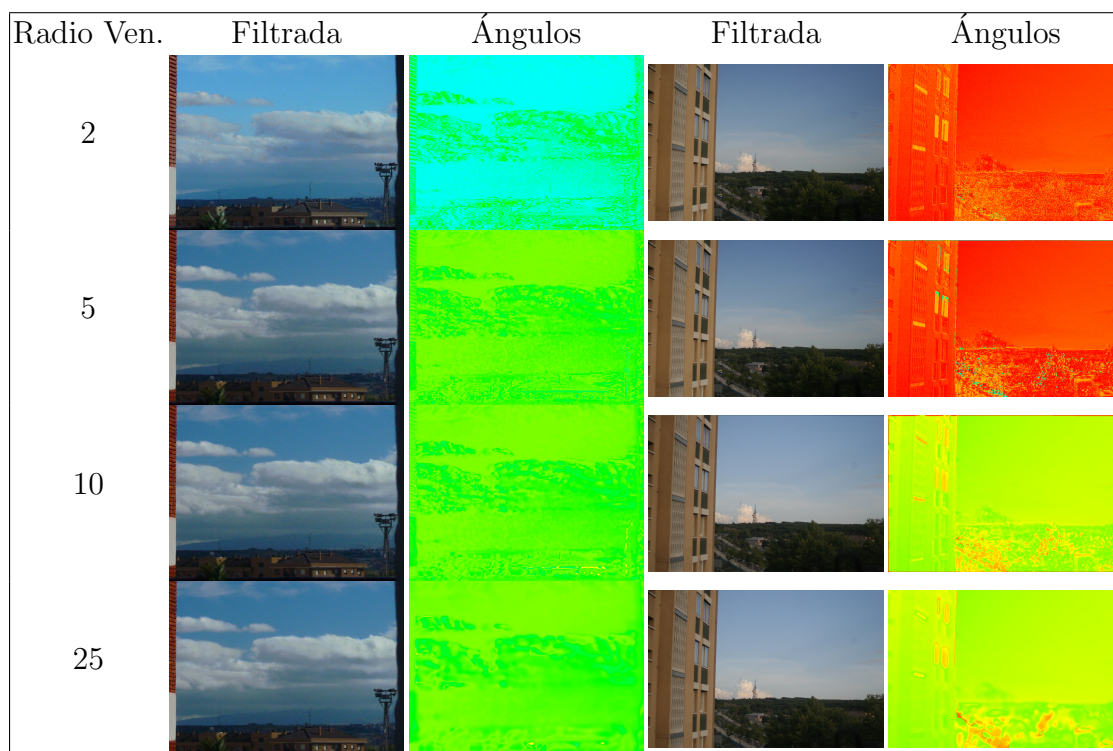


Figura 3.18: Comparación del impacto del tamaño de la ventana en los filtros de optimización del contraste.

En la Figura 3.18 se puede ver el efecto que tiene la selección de parámetros en el resultado. Si el tamaño es muy pequeño, el filtrado de cada píxel dependerá solo de los píxeles colindantes. Esto genera que cada píxel pueda tener un ángulo muy distinta a otros píxeles cercanos, por lo que puede dar lugar a resultados ruidosos en algunas zonas. Por el contrario, cuanto más grande sea la ventana, el filtrado de cada píxel dependerá de muchos más píxeles que en el caso anterior, por lo

que los cambios bruscos entre pixeles se limitan, generando resultados más suaves, aunque el ángulo de filtrado será menos específico de ese pixel. Este compromiso con el tamaño de la ventana hace que distintos tamaños puedan generar resultados muy distintos, como se puede observar en el salto entre radio 5 y radio 10 de la Figura 3.18. Por ejemplo, en la última columna de la Figura 3.18, se puede ver como en la zona arbolada de la imagen, para radio 2 y 5 el ángulo seleccionado varía mucho, mientras que para los tamaños mayores, el ángulo de filtrado se estabiliza.

Como se ha comentado, el tiempo de ejecución de este filtrado depende mucho de sus parámetros. Para radio=2, 20-30 segundos, para radio 5, entorno a 100 segundos, y el tiempo continua creciendo debido a que el coste de este filtrado es  $O(N * M * |W|)$ .

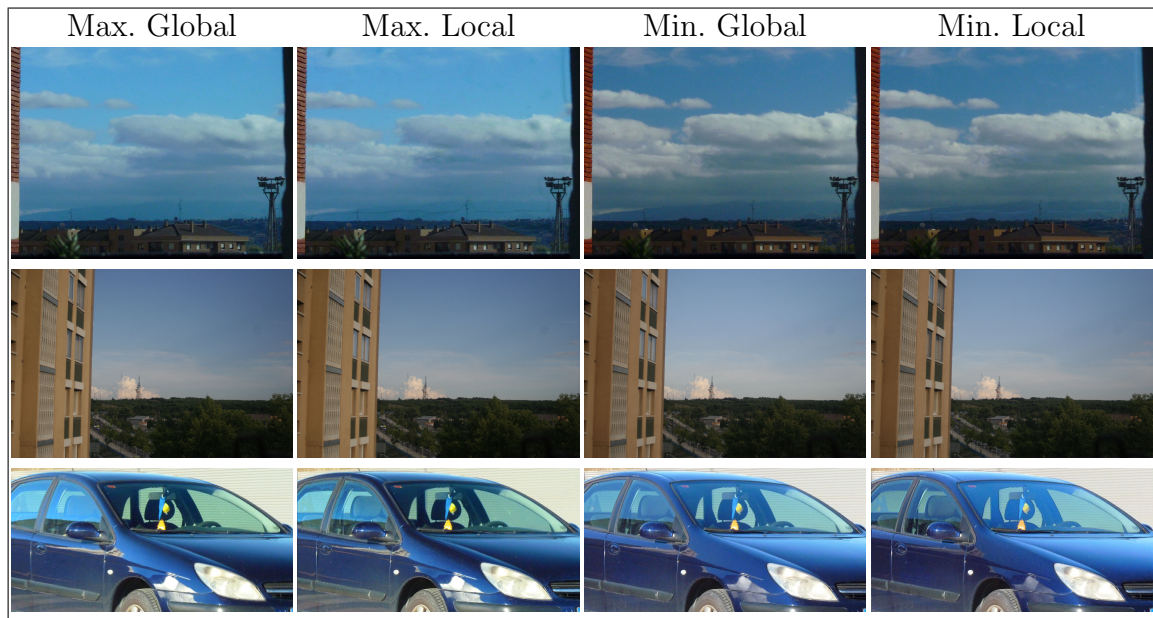


Figura 3.19: Comparativa para distintas escenas entre los filtros de optimización del contraste globales y locales

En la Figura 3.19 se puede ver una comparativa entre el filtrado local y el global. Si bien no se observan muchas diferencias entre ambos, ya que estas son sutiles, la introducción del filtrado local permitirá la utilización de estos para la herramienta por pinceles, que como se mostrara en el capítulo 4, se podrá utilizar para maximizar y minimizar el contraste en zonas concretas de la imagen, utilizando un pincel como ventana. Los filtros locales aplicados a toda la imagen requieren de mucho tiempo de ejecución para funcionar, y sin embargo el resultado no difiere mucho de los filtros globales en muchos casos. Por otro lado, una de las ventajas de los filtros locales es que permiten jugar con varios parámetros para conseguir distintos resultados.

## Optimización de la saturación

En el apartado global, se definía  $S(p)$  en la ecuación 3.20, y se utilizaba la suma de la saturación de cada pixel como saturación global. Para el filtrado local, se calcula para cada pixel de forma independiente el ángulo de polarización que maximice o minimice la saturación del color de ese pixel.

Los resultados obtenidos se pueden ver en la siguiente Figura 3.20:

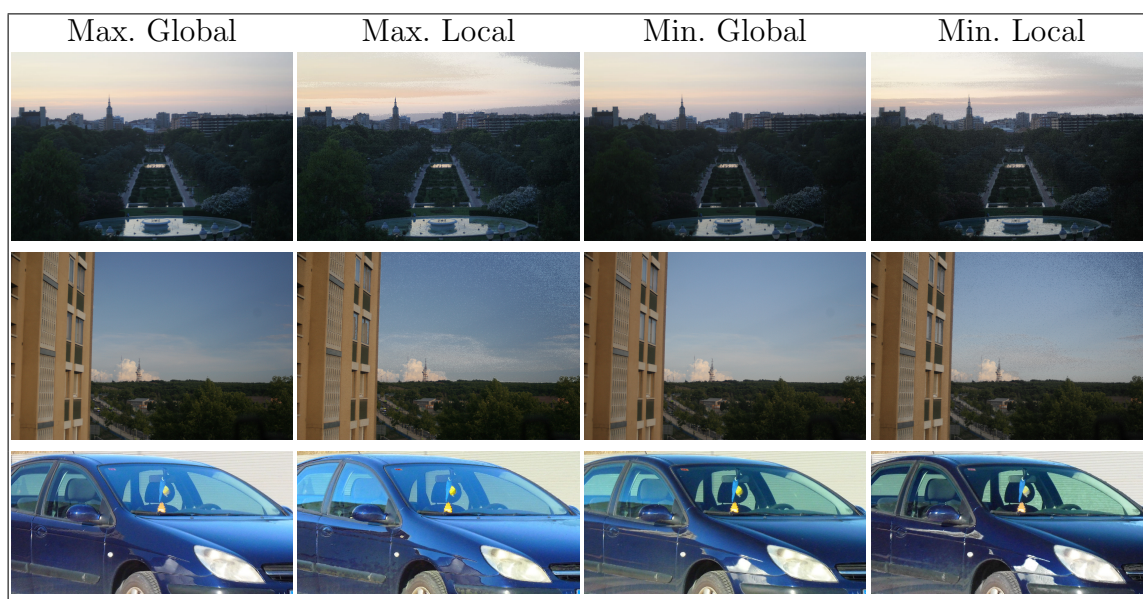


Figura 3.20: Comparativa para distintas escenas entre los filtros de optimización de la saturación globales y locales

En la Figura 3.20 se puede observar como el filtrado funciona, el mejor punto para observar este cambio es de nuevo el verde de los arboles o el cielo. Sin embargo, se puede observar como en el cielo hay **cambios bruscos** de un color a otro. Este se debe a que la saturación no es un parámetro que cambio linealmente con el ángulo de polarización. En la figura 3.15 se puede observar como al optimizar localmente la luminosidad, parámetro que si que cumple esta propiedad, el ángulo que se aplica en cada pixel varía gradualmente a lo largo de una superficie. Sin embargo, al observar el ángulo que se aplica en la optimización de la saturación no se observa este cambio gradual (ver Figura 3.21)



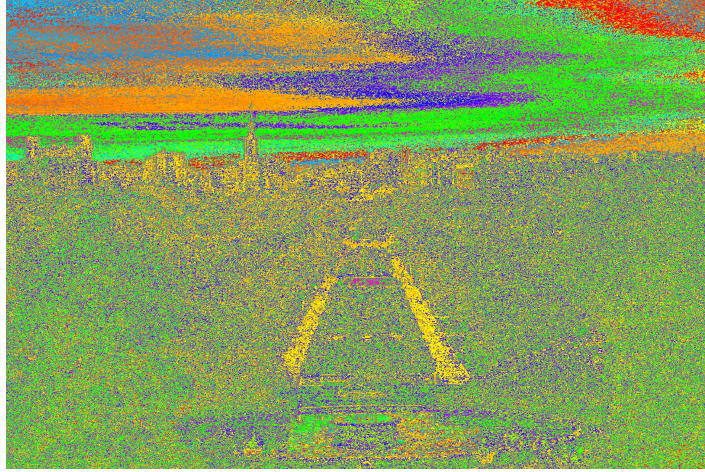


Figura 3.21: Ángulo seleccionado para cada pixel en la minimización local de la saturación

Al igual que sucedía con el contraste, matemáticamente, el objetivo se cumple, y los filtrados locales tienen un índice de saturación mejor que los globales. Sin embargo, los cambios bruscos generan imágenes artificiales.

# Capítulo 4

## Edición con pinceles

En el capítulo 3 se ha explicado como, gracias al filtrado en la fase de post-proceso, se pueden aplicar filtros distintos a distintas zonas de la imagen, algo imposible de conseguir mediante la fotografía convencional. Una de las formas de explotar esta ventaja, es desarrollar una herramienta mediante la cual se puedan aplicar distintos tipos de filtrado en zonas concretas de la imagen. Así como en el apartado 3.4.2 se conseguía maximizar o minimizar un parámetro en toda la imagen, el objetivo ahora es poder **aplicar filtros con objetivos distintos en diferentes zonas de la misma imagen**, permitiendo maximizar un parámetro en un área y minimizarlo en otra.



Figura 4.1: Frames donde se puede apreciar la aplicación progresiva de un pincel para minimizar el brillo.

Para ello se ha decidido desarrollar una herramienta que permita esta funcionalidad mediante el **uso de pinceles**. Usando un pincel circular se puede maximizar o minimizar un parámetro concreto en cualquier zona de la imagen. El pincel funciona aplicando el filtro correspondiente con un degradado gaussiano desde el centro del pincel hacia sus extremos. Esta herramienta permite modificar los tres parámetros analizados en el capítulo anterior, utilizando dicho pincel, y guardar la imagen

modificada. Además de estos tres parámetros, se ha añadido un pincel que haga la función de goma, para restablecer la imagen original y eliminar un filtro. La interfaz es simple, constando de una ventana con la imagen y el pincel (ver Figura 4.1). Los pinceles se seleccionan mediante atajos de teclado, y se utiliza la terminal para introducir los parámetros necesarios.

Si bien los filtros locales mostrados 3.4.2 eran rápidos en su ejecución para toda la imagen, al aplicarlos en tiempo real el resultado era una aplicación algo lenta, que para pinceles grandes en los parámetros lentos, como el contraste y la saturación, solo conseguía una tasa de *frames* por segundo de 15-20. Los **filtros locales son fácilmente paralelizables**, ya que cada pixel recibe un procesamiento distinto. Aprovechando esta propiedad, se han implementado estos algoritmos para que utilicen varios hilos para procesar los pixeles correspondientes del pincel. El número de hilos se determina en función del radio, de forma que no se utilicen demasiados hilos para pinceles pequeños ni viceversa, pocos hilos para pinceles grandes que si los necesitan. Con este cambio, se obtienen *frame rates* de más de 60 para pinceles pequeños (de un radio de unos 20-30 pixeles), entre 30 y 60 para pinceles medianos (30-80 pixeles) y para pinceles grandes empiezan a decaer progresivamente hasta 15-20 fps. Esto ocurre en los pinceles de saturación y contraste, debido a la necesidad de resolver una ecuación de grado 4. Por el contrario, los pinceles del brillo se mantienen siempre por encima de 30 fps aun con pinceles de más de 200 pixeles de radio. Estos resultados corresponden a un equipo con un procesador i5-7600 con 16GB de RAM.

A continuación se van a mostrar distintos resultados en los que se modifican los parámetros presentados. En primer lugar, se van a mostrar resultados donde se vea clara la diferencia entre maximizar o minimizar un parámetro en una imagen, y posteriormente, modificando más de un parámetro en la imagen.

### Pincel de luminosidad

El pincel de brillo, al igual que sus correspondientes filtros locales, es el que más impacto visual genera. Este pincel permite, entre otras cosas, eliminar reflejos de las superficies o potenciarlos. La herramienta por pinceles tiene varias ventajas respecto a la aplicación de filtros locales. La más obvia es que permite maximizar o minimizar el brillo en zonas distintas de la imagen. En la Figura 3.16 se podía observar como los filtros globales no permitían maximizar o minimizar la luminosidad en dos superficies distintas, mientras que los filtros locales permiten realizar esa acción en cualquier superficie. El único inconveniente es que no se puede **maximizar una zona**

y **minimizar otra**. Por ejemplo, usando la misma escena que en la Figura 3.16, no es posible maximizar la luminosidad de la pantalla y minimizar la de la carátula, sin embargo, con la introducción de la herramienta de pinceles, esto se puede realizar.

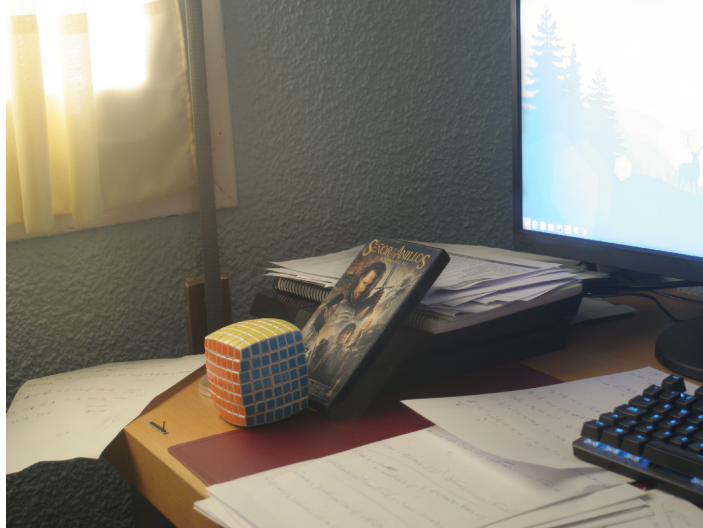


Figura 4.2: Ejemplo del uso de la herramienta de pinceles para la luminosidad

Además de esta ventaja principal, existen otras. Como se ha comentado en el apartado de la captura de imágenes(ver Sección 3.2), la aplicación de un filtro de polarización era una combinación de los parámetros de Stokes de una escena. Esto generaba problemas como la aparición de "fantasmas"(ver Figura 3.4). Si en alguna imagen había objetos móviles, estos aparecerían borrosos o traslucidos en la imagen filtrada. A continuación se va a mostrar la maximización del brillo en una imagen que sufre este problema:



Figura 4.3: Ejemplo del problema del movimiento entre distintas capturas

El objetivo principal en esta imagen es maximizar los brillos en la superficie del coche principal de la imagen. Sin embargo, debido a que el coche esta en la calle, en las

diferentes capturas para generar la imagen de Stokes ha habido movimiento de otros vehículos al fondo de la imagen, haciendo que esa parte de la imagen quede mal. Si en lugar de aplicar un filtro local, se utiliza la herramienta de pinceles, se puede maximizar solo la superficie del coche objetivo, evitando así que en el fondo de la imagen se vean coches superpuestos y translucidos dando lugar al resultado deseado.



Figura 4.4: Utilización de la herramienta de pinceles para evitar el problema de la Figura 4.3

En esta última figura se puede observar como en los coches del fondo se pueden ver sin ningún tipo de distorsión, mientras que en la Figura 4.3, se pueden observar coches translucidos y algunos colores que no deberían aparecer, como el azul sobre las puertas del coche blanco.

### **Pincel de contraste**

El pincel de contraste permite maximizar o minimizar el contraste en distintas zonas de la imagen. Utilizando la misma imagen de ejemplo que en los filtros locales, la Figura 4.5, se podría por ejemplo maximizar el contraste del cielo y minimizar el de los edificios. El resultado es el siguiente:





Figura 4.5: Ejemplo del uso de la herramienta de pinceles para el contraste

### **Pincel de saturación**

Para el pincel de saturación se va a utilizar también la misma imagen que en los filtros locales, 4.6. Se comentaba que en ese caso, el matiz que mas se diferenciaba era el verde de los arboles al maximizar y minimizar la saturación. En este ejemplo se va a maximizar la saturación en la parte de la izquierda, mientras que se va a minimizar la parte de la derecha. El resultado queda:



Figura 4.6: Ejemplo del uso de la herramienta de pinceles para la saturación

Al igual que ocurría con el filtrado por pixel, dependiendo de la escena se pueden obtener resultados artificiales con algunos cambios bruscos en el color, como se ha visto en la Figura 3.20.

## Utilización de varios pinceles en una imagen

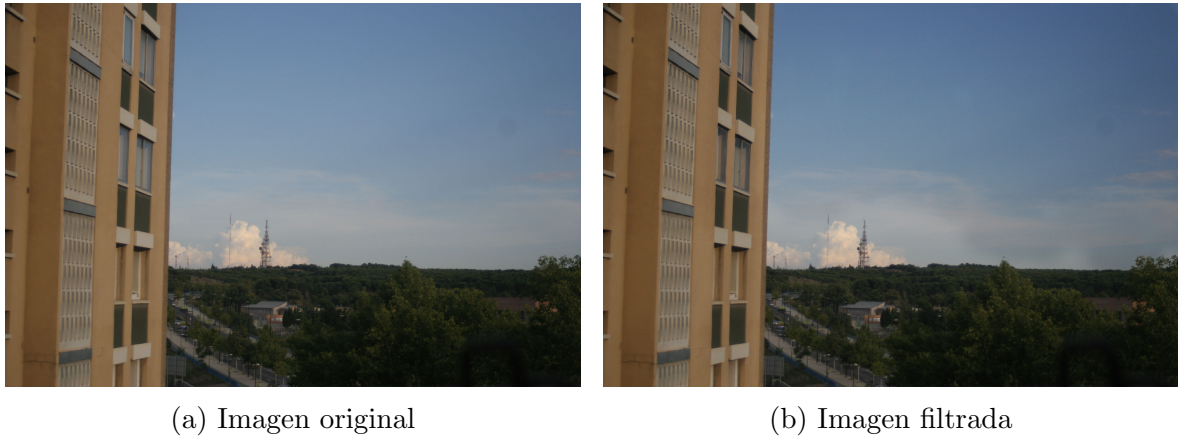


Figura 4.7: Ejemplo del uso de varios pinceles

En la Figura 4.7 se puede observar como, se ha minimizado el brillo en los cristales de las ventanas de la parte de abajo, mientras que se ha maximizado en las ventanas de la parte superior. Además se ha maximizado el contraste del cielo.

Si bien el pincel del brillo funciona a la perfección independientemente de la resolución de la imagen y del tamaño del pincel, para el contraste y la saturación se empiezan a apreciar algunas ralentizaciones cuando el pincel representa un tamaño bastante grande respecto a la imagen, y esta además esta en alta resolución. Esto se debe a que para calcular estos filtros es necesario obtener las raíces de un polinomio de grado cuatro, proceso que aunque se realiza con un método simplificado, requiere de bastantes más cálculos que una simple arcotangente.

Esta herramienta representa una mejora respecto a los filtros locales, ya que permiten mayor control artístico por parte del usuario

# Capítulo 5

## Conclusiones

A continuación se van a valorar los objetivos propuestos en la introducción. Se ha realizado un estudio del comportamiento de la luz polarizada, así como su representación mediante los vectores de Stokes y la manipulación de estos mediante las matrices de Mueller. Posteriormente, se ha introducido un **método de captura de imágenes de Stokes** sencillo y barato, junto con un formato para almacenarlas, que ha permitido obtener un banco de pruebas para la aplicación. Utilizando estas imágenes se ha desarrollado una aplicación capaz de aplicar los filtros polarizadores básicos de la fotografía, tanto el polarizador lineal como el circular, introduciendo además un filtrado parametrizado que permite obtener filtros imposibles de reproducir en fotografía. Se han introducido filtros capaces de **seleccionar automáticamente el mejor filtro** lineal para optimizar los parámetros comentados, la luminosidad, el contraste y la saturación. Con ellos, se han introducido los **filtros locales**, que son capaces de aplicar un filtrado distinto a diferentes zona de la imagen, consiguiendo resultados imposibles de generar mediante la fotografía convencional. Además, se ha mostrado la posibilidad de utilizar estos filtros en tiempo real, utilizándolos en una aplicación que permite editar imágenes con pinceles, consiguiendo de nuevo generar **imágenes irreproducibles con la fotografía**.

Si bien se ha conseguido realizar la optimización de los distintos parámetros, cabe destacar el **buen funcionamiento de los filtros de optimización de la luminosidad**, que entre otras cosas, permiten maximizar y minimizar los reflejos de una imagen de una forma rápida, consiguiendo unos resultados muy positivos, llegando a eliminar completamente los reflejos en algunas superficies, ver a través de los cristales de un coche o hacer visibles reflejos que sin este filtrado pasarían desapercibidos. Por otro lado, los resultados del contraste y la saturación no han sido tan positivos. Los filtros globales si que han conseguido resultados positivos, mientras que los locales han dado resultados algo más agri dulces. Para el contraste, el problema de depender de

una ventana a incrementado notablemente el coste de los algoritmos. Por otro lado, al introducir esta ventana, se han introducido nuevos parámetros con los que trabajar para obtener distintos resultados, además de permitir su utilización en la herramienta de pinceles. En la saturación, se han encontrado de nuevo problemas con el filtrado local, haciendo que en algunas situaciones generen problemas de ruido generando imágenes artificiales, mientras que en su aplicación por pinceles se han conseguido resultados positivos.

## **Trabajo futuro**

En cuanto a los puntos a explorar en un futuro, el punto más importante a mejorar es la captura de los parámetros de Stokes. Si bien se trata de un método muy simple y barato, reproducible con poco presupuesto ya que solo precisa de una cámara y dos filtros polarizadores, este método tiene numerosas limitaciones, como se trata de la necesidad de tomar 6 imágenes distintas para calcular estos parámetros. Esto provoca que no se puedan obtener los parámetros de Stokes de escenas con objetos no estáticos, generando resultados no deseados. En este campo ya se han desarrollado varios métodos que permiten capturar los parámetros de Stokes en una sola toma. Esto abre la puerta, ya no solo a la utilización de estos métodos para aplicar los mismos filtros que se han presentado en este TFG para la edición de imágenes, sino que también permitiría trabajar con vídeo capturando los parámetros de Stokes, donde se podrían utilizar los filtros presentados en este trabajo. Esto podría tener aplicaciones en seguridad, utilizándolo en cámaras que permitirían por ejemplo, ver a través de cristalerías o ventanas de coche, pudiendo identificar a personas tras ellos.

Otro claro punto de mejora es la herramienta de pinceles, ya que si bien cumple su función y ha permitido demostrar que los filtros son aplicables en tiempo real, se trata de una interfaz muy limitada, donde algunas tareas son algo tediosas de realizar. Además funciona en su totalidad con atajos de teclado que el usuario tendría que memorizar. Sin embargo, los filtros locales son funcionales a la perfección en su aplicación por pinceles, con lo que se podría desarrollar una interfaz más avanzada, similar a lo que podrían ser programas como GIMP o Photoshop.

Además de estos puntos, también se podría aplicar la metodología mostrada a otro tipo de parámetros y valores de una imagen. En este trabajo se han mostrado los resultados para la luminosidad, el contraste y la saturación, y de la misma forma que se han optimizado estos parámetros, se podría aplicar sobre otros parámetros.

# Bibliografía

- [1] G. G. Stokes. On the Composition and Resolution of Streams of Polarized Light from different Sources. *Transactions of the Cambridge Philosophical Society*, 9:399, 1851.
- [2] Hans Mueller. Memorandum on the polarization optics of the photoelastic shutter. *Report of the OSRD project OEMsr-576*, 2, 1943.
- [3] David Osante Sicilia and Adolfo Muñoz Orbañanos. Fotografía computacional para el filtrado de la polarización en fase de post-proceso. Trabajo fin de máster, Universidad de Zaragoza, 2013.
- [4] Fanchao Yang, Xingjia Tang, Bingliang Hu, Ruyi Wei, Liang Kong, and Yong Li. A method of removing reflected highlight on images based on polarimetric imaging. *Journal of Sensors*, 2016:7, 2016.
- [5] Gang Yao and Lihong V. Wang. Two-dimensional depth-resolved mueller matrix characterization of biological tissue by optical coherence tomography. *Opt. Lett.*, 24(8):537–539, Apr 1999.
- [6] Xingzhou Tu, Oliver J. Spies, Xiaobo Tian, Neal Brock, Rongguang Liang, and Stanley Pau. Division of amplitude rgb full-stokes camera using micro-polarizer arrays. *Opt. Express*, 25(26):33160–33175, Dec 2017.
- [7] [https://en.wikipedia.org/wiki/Relative\\_luminance](https://en.wikipedia.org/wiki/Relative_luminance).
- [8] <https://github.com/sasamil/Quartic>.
- [9] George W. Kattawar and Charles N. Adams. Stokes vector calculations of the submarine light field in an atmosphere-ocean with scattering according to a rayleigh phase matrix: Effect of interface refractive index on radiance and polarization. *Limnology and Oceanography*, 34(8):1453–1472, 1989.
- [10] Grupo de Innovación Docente en Óptica Física y Fotónica. Curso de Óptica. <http://www.ub.edu/javaoptics/teoria/castella/node8.html>.

- [11] R. Espinosa-Luna and S. Hinojosa-Ruíz and G. Atondo-Rubio. Manejo e interpretación polarimétrica de las matrices de mueller. *Revista mexicana de física*, 55, 2009.
- [12] Russell A Chipman. Polarimetry. In Michael Bass, Casimer Decusatis, Jay Enoch, Vasudevan Lakshminarayanan, Guifang Li, Carolyn Macdonald, Virendra Mahajan, and Eric Van Stryland, editors, *Handbook of optics*, chapter 22. 2001.
- [13] J. A. North and M. J. Duggin. Stokes vector imaging of the polarized sky-dome. *Appl. Opt.*, 36(3):723–730, Jan 1997.

# Lista de Figuras

1.1. Distintas tecnologías que permiten una edición a posteriori . . . . .	1
1.2. Comparativa de la misma fotografía tomada con y sin polarizador donde se observan los efectos que se pueden conseguir con estos. . . . .	2
1.3. Esquema de las analogías entre las tecnologías HDR, lytro y el objetivo a conseguir mediante la representación de Stokes. . . . .	3
2.1. Polarización de una onda. La onda no polarizada oscila en todas las direcciones posibles del plano de polarización, al atravesar el elemento polarizador, oscila solo en una dirección. . . . .	6
2.2. Ilustración de los diferentes tipos de polarización en función de la forma que generan en el plano de polarización. . . . .	8
2.3. Cambios en la elipse de polarización en función de la diferencia de fase $\delta$	9
2.4. Elipse de polarización sobre los ejes de referencia x e y. . . . .	9
2.5. Polarizador lineal: la onda oscila solo en una dirección tras atravesar este elemento polarizador. . . . .	13
2.6. Cambio de fase de $180^\circ$ . . . . .	14
2.7. Ejes del plano de polarización rotados $\theta$ grados . . . . .	15
3.1. Visualización de los parámetros de Stokes de una escena utilizando la normalización presentada en la ecuación 3.1. . . . .	18
3.2. Capturas de 2 escenas distintas capturadas mediante el método presentado en [3] . . . . .	19
3.3. Capturas de 3 escenas distintas capturadas mediante el método presentado en la Sección 3.2 . . . . .	20
3.4. Parámetro I de una imagen de Stokes, generada a partir de dos imágenes en las que un objeto aparece en una captura y no en la otra. . . . .	20
3.5. Ejemplo del problema de alineación . . . . .	21
3.6. Imagen filtrada con un polarizador lineal a $75^\circ$ . . . . .	22
3.7. Imagen filtrada con un polarizador circular a derechas . . . . .	23

3.8. Imagen filtrada con un polarizador por parámetros con $c1=1$ , $c2=0$ y $c3=-1$ . . . . .	24
3.9. Resultados al aplicar distintos filtros(El filtrado por parámetros corresponde a los valores $c1=1$ , $c2=-1$ , $c3=-1$ ) . . . . .	24
3.10. Resultado de aplicar el filtro de optimización del brillo global. En la maximización se ha aplicado un filtro lineal a 130.22 grados, mientras que en la minimización uno a 40.22 grados. . . . .	28
3.11. Resultados de aplicar los filtros de optimización globales del contraste. Se puede observar el principal cambio en el cielo. . . . .	30
3.12. Resultados de aplicar los filtros de optimización globales de la saturación. Se pueden observar los cambios en el cielo y en la vegetación. . . . .	31
3.13. Detalle de una imagen de la Figura 4.6 en la optimización de la saturación donde se observa un cambio sutil en el color de los arboles .	32
3.14. Comparativa para distintas escenas entre los filtros de optimización del brillo globales y locales . . . . .	33
3.15. Ángulo seleccionado para cada pixel en la maximización local de la luminosidad . . . . .	34
3.16. Comparativa entre la optimización global y local de la luminosidad en una escena con una fuente de luz polarizada. . . . .	35
3.17. Valores de una ventana gaussiana $5 \times 5$ con $\sigma = 1,5$ . . . . .	36
3.18. Comparación del impacto del tamaño de la ventana en los filtros de optimización del contraste. . . . .	37
3.19. Comparativa para distintas escenas entre los filtros de optimización del contraste globales y locales . . . . .	38
3.20. Comparativa para distintas escenas entre los filtros de optimización de la saturación globales y locales . . . . .	39
3.21. Ángulo seleccionado para cada pixel en la minimización local de la saturación . . . . .	40
4.1. Frames donde se puede apreciar la aplicación progresiva de un pincel para minimizar el brillo. . . . .	41
4.2. Ejemplo del uso de la herramienta de pinceles para la luminosidad . . .	43
4.3. Ejemplo del problema del movimiento entre distintas capturas . . . . .	43
4.4. Utilización de la herramienta de pinceles para evitar el problema de la Figura 4.3 . . . . .	44
4.5. Ejemplo del uso de la herramienta de pinceles para el contraste . . . . .	45
4.6. Ejemplo del uso de la herramienta de pinceles para la saturación . . . .	45



4.7. Ejemplo del uso de varios pinceles . . . . .	46
A.1. Elipse de polarización en función de $\delta$ . . . . .	56
B.1. Polarizador lineal . . . . .	58
B.2. Retardador . . . . .	60